

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

**ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ  
КАФЕДРА ПРОМИСЛОВОЇ ЕЛЕКТРОНІКИ**

«На правах рукопису»  
УДК 338.2

« До захисту допущено»

авідувач кафедри

\_\_\_\_\_ Ю.С. Ямненко  
(підпис) (ініціали, прізвище)

“ \_\_\_\_\_ ” \_\_\_\_\_ 2018р.

**Магістерська дисертація**

**на здобуття ступеня магістра**

зі спеціальності \_\_\_\_\_ 171 Електроніка  
(код і назва)

спеціалізації \_\_\_\_\_ Електронні компоненти і системи

на тему: Оцінка режимів MicroGrid на базі регресійного аналізу

Виконав: студент \_\_\_\_\_ II \_\_\_\_\_ курсу, групи \_\_\_\_\_ ДС-71мп  
(шифр групи)

Орлов Євгеній Андрійович  
(прізвище, ім'я, по батькові)

\_\_\_\_\_ (підпис)

Науковий керівник д.т.н., проф. Ямненко Ю.С.  
(посада, науковий ступінь, вчене звання, прізвище та ініціали)

\_\_\_\_\_ (підпис)

Консультант \_\_\_\_\_  
(назва розділу) (науковий ступінь, вчене звання, прізвище, ініціали)

\_\_\_\_\_ (підпис)

Рецензент доктор технічних наук, професор Тимофєєв В.І.  
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали)

\_\_\_\_\_ (підпис)

Засвідчую, що у цій магістерській  
дисертації немає запозичень з праць  
інших авторів без відповідних посилань.

Студент \_\_\_\_\_  
(підпис)

Київ – 2018 року

**Національний технічний університет України  
“Київський політехнічний інститут  
імені Ігоря Сікорського”**

Факультет електроніки  
( повна назва )

Кафедра промислової електроніки  
( повна назва )

Рівень вищої освіти – другий (магістерський) за освітньо - професійною програмою

Спеціальність 171 Електроніка  
(шифр і назва)

Спеціалізація Електронні компоненти і системи

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_  
(підпис)                      Ю.С. Ямненко  
( прізвище ініціали )

«03» вересня 2018 року

**З А В Д А Н Н Я  
НА МАГІСТЕРСЬКУ ДИСЕРТАЦІЮ СТУДЕНТУ**

Орлову Євгенію Андрійовичу  
(прізвище, ім'я, по батькові)

1. Тема проекту «Оцінка режимів MicroGrid на базі регресійного аналізу»  
науковий керівник дисертації Ямненко Юлія Сергіївна, д.т.н., проф.  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від « 07 » листопада 2018 року № 4114-с

2. Строк подання студентом проекту 6-го грудня 2018 року

3. Об'єкт дослідження Процеси генерації та споживання електричної енергії у MicroGrid

4. Предмет дослідження (вихідні дані для магістерської дисертації за освітньо-професійною програмою) Розробка методів оцінювання ефективності режимів роботи MicroGrid на базі регресійного аналізу

5. Перелік завдань, які потрібно розробити 1. Оцінка можливості прогнозування надлишку електричної енергії за допомогою регресійного аналізу; 2. Розробка методу обробки часових рядів параметрів режимів MicroGrid за допомогою регресійного аналізу; 3. Розробка системи збору, обробки та передачі даних; 4. Вибір та обґрунтування типу бази даних; 5. Розробка програмного забезпечення для оцінки режимів на базі регресійного аналізу.

6. Перелік графічного (ілюстративного) матеріалу Слайди презентації
7. Орієнтовний перелік публікацій 1. Орлов Є. А. Пристрої поновлених джерел енергії у системі Micro Grid та Smart House // Збірник тез доповідей V міжнародної науково-технічної конференції молодих учених та студентів «Актуальні задачі сучасних технологій» 17-18 листопада 2016 року, Тернопіль – 2016. С. 182 -183 // <http://tntu.edu.ua/storage/news/00002831/zbirnyk-tom2.pdf>;  
2. Орлов Є. А. оцінка ефективного використання та продажу електроенергії отриманої від відновлювальних джерел енергії в системі Microgrid // Вчені записки таврійського національного університету імені в. І. Вернадського серія: технічні науки, том 29 (68) №5 2018, частина 2, Київ – 2018. – с. 26-30 \_\_

#### 8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

#### 9. Дата видачі завдання «03» вересня 2018 року

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання магістерської дисертації	Строки виконання етапів магістерської дисертації	Примітка
1	Огляд літератури	1.09.2018-20.09.2018	виконано
2	Аналіз існуючих методів регресійного аналізу та спосіб використання методу на практиці та в електронних системах.	21.09.2018-10.10.2018	виконано
3	Розробка тестового макету пристрою збору та передачі інформації	11.10.2018-24.10.2018	виконано
4	Розробка програмного забезпечення для отримання кількості залишкової електроенергії в MicroGrid	28.10.2018-28.11.2018	виконано
5	Розробка стартап-проекту	29.11.2018 – 04.12.2018	виконано

Студент

\_\_\_\_\_  
(підпис)

Орлов Є.А.  
(ініціали, прізвище)

Науковий керівник дисертації

\_\_\_\_\_  
(підпис)

Ямненко Ю.С.  
(ініціали, прізвище)

## АНОТАЦІЯ

У магістерській дисертації розроблено систему, яка виконує функції збору даних про споживання та генерацію електричної енергії від пристроїв системи розподіленої генерації MicroGrid. Накопичені дані зберігаються у базі даних та підлягають обробці на базі методів кореляційно-регресійного аналізу, що дає можливість обчислити прогнозований обсяг залишкової енергії, виробленої альтернативними генераторами, яка підлягає реалізації за «зеленим тарифом» на локальному ринку електроенергії. Представлено дослідження взаємодії факторів, що впливають на генерацію, споживання та продаж електричної енергії з використанням регресійного аналізу. Виявлено основні показники, які мають найбільший вплив на кількість залишкової енергії та зроблені висновки з приводу ефективного використання цього методу для локального ринку MicroGrid. Розроблено програмне забезпечення для проведення лінійного регресійного аналізу даних.

**Ключові слова:** системи розподіленої генерації, альтернативні джерела, MicroGrid, регресійний аналіз, кореляція, пристрої збору та передачі інформації, бази даних.

## АННОТАЦИЯ

В магистерской диссертации разработана система, которая выполняет функции сбора данных о потреблении и генерацию электрической энергии от устройств системы распределенной генерации MicroGrid. Накопленные данные хранятся в базе данных и подлежат обработке на базе методов корреляционно-регрессионного анализа, что позволяет вычислить прогнозируемый объем остаточной энергии, производимой альтернативными генераторами, подлежащей реализации по «зеленому тарифу» на локальном рынке электроэнергии. Представлены исследования взаимодействия факторов,

влияющих на генерацию, потребление и продажи электрической энергии с использованием регрессионного анализа. Выявлены основные показатели, которые оказывают наибольшее влияние на количество остаточной энергии и сделаны выводы по поводу эффективного использования этого метода для локального рынка MicroGrid. Разработано программное обеспечение для проведения линейного регрессионного анализа данных.

**Ключевые слова:** системы распределенной генерации, альтернативные источники, MicroGrid, регрессионный анализ, корреляция, устройства сбора и передачи информации, базы данных.

## ANNOTATION

The master's thesis developed a system that performs the functions of collecting data on the consumption and generation of electrical energy from devices of the distributed generation system MicroGrid. The accumulated data is stored in a database and subject to processing based on the methods of correlation and regression analysis, which allows us to calculate the predicted amount of residual energy produced by alternative generators to be sold at a “green tariff” in the local electricity market. Studies are presented on the interaction of factors affecting the generation, consumption and sale of electrical energy using regression analysis. The main indicators that have the greatest impact on the amount of residual energy are identified and conclusions are drawn about the effective use of this method for the local MicroGrid market. Developed software for linear regression data analysis.

**Keywords:** distributed generation systems, alternative sources, MicroGrid, regression analysis, correlation, information collection and transmission devices, databases.

## ЗМІСТ

ВСТУП	7
1. РОЗДІЛ 1. МЕТОДИ ПРОГНОЗУВАННЯ ЕНЕРГОЕФЕКТИВНОСТІ В СИСТЕМІ MICROGRID	10
1.1. Области застосування MicroGrid	11
1.2. Інформаційна система моніторингу та аналізу моделі локальної активно-адаптивної мережі	14
1.3. Програмно-технічний комплекс оптимізації MicroGrid	17
1.3.1. Довгострокова оптимізація	19
1.3.2. Короткострокова оптимізація	20
1.4. Фактографічні, експертні і комбіновані методи прогнозування	22
1.5. Моделі часових рядів	24
1.6. Переваги і недоліки основних моделей прогнозування	26
Висновки до першого розділу	27
РОЗДІЛ 2. РЕГРЕСІЙНИЙ АНАЛІЗ	29
2.1. Основні поняття	29
2.2. Оцінка тісноти, суттєвості й лінійності (нелінійності) зв'язку між змінними	38
2.3. Оцінка ефективного використання та продажу електроенергії, отриманої від відновлювальних джерел енергії в системі MicroGrid	43
2.4. Лінійна регресія в програмному забезпеченні на мові C#	49
Висновки до другого розділу	57
РОЗДІЛ 3. БАЗИ ДАНИХ ТА СИСТЕМИ КЕРУВАННЯ БАЗАМИ ДАНИХ	59
3.1. Реляційні бази даних	59
3.2. Теорема ACID	62
3.3. Нереляційні бази даних	64
3.4. Порівняння SQL та NoSQL БД	67
3.5. Система керування базами даних MS SQL	69

3.6. Модель для MS SQL Server	71
Висновки до третього розділу	74
РОЗДІЛ 4. ПРИСТРІЙ ЗБОРУ, ОБРОБКИ ТА ПЕРЕДАЧІ ДАНИХ	75
4.1. Пристрої фіксації електричної енергії	75
4.2. Open Data Protocol	80
Висновки до четвертого розділу	82
РОЗДІЛ 5. РОЗРОБКА СТАРТАП – ПРОЕКТУ	83
5.1. Опис ідеї проекту	85
5.2. Технологічний аудит ідеї проекту	86
5.3. Аналіз ринкових можливостей запуску стартап-проекту	87
5.4. Розроблення маркетингової програми стартап-проекту	90
Висновки до п'ятого розділу	93
ВИСНОВКИ	94
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	95
ABSTRACT	101
Додаток	104

## ВСТУП

**Актуальність.** Сучасні дослідження, спрямовані на підвищення ефективності використання електричної енергії у системах розподіленої генерації MicroGrid та перерозподіл енергетичних потоків на локальних віртуальних ринках, підкреслюють актуальність задач моніторингу, оцінки та прогнозування обсягів генерації енергії. З точки зору сумісного розглядання технічних та економічних факторів функціонування MicroGrid, враховуються рівень попиту, ціна на електроенергію, «зелений» тариф, сезонність, надлишок енергії від альтернативних джерел. На підставі аналізу цих та інших показників формуються заявки на продаж та придбання енергії у гравців віртуального ринку.

**Мета і завдання досліджень:** Метою даної роботи є розробка методу прогнозування кількості електроенергії, яка може бути підготовлена для реалізації на локальному ринку MicroGrid, з урахуванням підвищення або зниження цін на зелений тариф та електроенергію в мережі. При цьому на базі кореляційно-регресійного аналізу часових рядів, які попередньо підлягають екстраполяційній обробці, встановлюється статистичний взаємозв'язок між параметрами і оцінюється вплив обраних параметрів на значення попиту на електричну енергію в майбутньому на прогнозований період.

Для досягнення мети розглядалися наступні задачі:

- Оцінка можливості прогнозування надлишку електричної енергії за допомогою регресійного аналізу;
- Розробка методу обробки часових рядів параметрів режимів MicroGrid за допомогою регресійного аналізу;
- Розробка системи збору, обробки та передачі даних;
- Вибір та обґрунтування типу бази даних;
- Розробка програмного забезпечення для оцінки режимів на базі регресійного аналізу.



### **Зв'язок роботи з науковими програмами, планами, темами.**

Робота виконана в Національному технічному університеті України «Київський політехнічний інститут» на кафедрі промислової електроніки в рамках ініціативної науково-дослідницької роботи «Підвищення показників енергоефективності та ресурсозбереження засобами силової електроніки для технології отримання високонадійних зварюваних з'єднань різнорідних матеріалів», № державної реєстрації 0116U006924.

**Об'єктом дослідження** є процеси генерації та споживання електричної енергії у MicroGrid.

**Предметом дослідження** є розробка методів оцінювання ефективності режимів роботи MicroGrid на базі регресійного аналізу.

**Методи досліджень.** При розв'язанні поставлених у роботі завдань був використаний методи Machine Learning та регресійного аналізу. Математичні та програмні розрахунки виконані на персональному комп'ютері з використанням пакету розробника «Visual Studio 2017» та миви програмування C#.

### **Наукова новизна даних досліджень полягає в наступному:**

- Отримала подальший розвиток теорія дослідження та аналізу режимів роботи систем розподіленої генерації;
- На основі регресійного аналізу було розроблено систему для знаходження оцінки ефективного використання електроенергії для Microgrid з врахуванням сезонних параметрів та даних споживання енергії пристроями системи.

### **Практичне значення отриманих результатів:**

- Розроблений експериментальний макет дозволяє реалізувати збір, обробку та передачу даних за допомогою стандартних протоколів до бази даних;
- З використанням мови програмування C# розроблено програмне забезпечення, що здійснює розрахунок надлишку виробленої енергії на базі регресійного аналізу.

**Особистий внесок здобувача.** Магістерська робота є узагальненням результатів теоретичних та експериментальних досліджень, проведених автором з консультаціями наукового керівника. У роботах опублікованих із співавторами, автору у [1] належить пошук та аналіз інформації щодо існуючих програмних продуктів для моделювання систем електроживлення MicroGrid, редагування, оформлення; У [2] пошук та аналіз інформації щодо методів машинного навчання для знаходження залишку електричної енергії для подальшої реалізації на локальному ринку MicroGrid, редагування, оформлення, розробка пристрою збору даних та програмного забезпечення.

Апробація, Результати дисертаційних досліджень доповідалися та обговорювалися на V науково-технічній конференції молодих учених та студентів «Актуальні задачі сучасних технологій» 17-18 листопада 2016 року, м. Тернопіль .

**Публікації.** Основні положення і результати, отримані в дисертаційній роботі, були описані в наступних статтях:

- Орлов Є. А. Пристрої поновлених джерел енергії у системі micro grid та smart house // Збірник тез доповідей V міжнародної науково-технічної конференції молодих учених та студентів «Актуальні задачі сучасних технологій». 17-18 листопада 2016 року, Тернопіль – 2016. Том 2, С. 182 -183 // <http://tntu.edu.ua/storage/news/00002831/zbirnyk-tom2.pdf>;
- Орлов Є. А. Оцінка ефективного використання та продажу електроенергії отриманої від відновлювальних джерел енергії в системі Microgrid // Вчені записки таврійського національного університету імені в. І. Вернадського серія: технічні науки, том 29 (68) №5 2018, частина 2, кийв – 2018. – с. 26-30.

**Структура та обсяг дисертації.** Дисертація складається зі вступу, п'яти розділів, висновків, списку використаних джерел із 56 найменувань Загальний обсяг дисертаційної роботи становить 122 сторінок, у тому числі 90 сторінок основного тексту 15 рисунків та 17 таблиць.

## РОЗДІЛ 1. МЕТОДИ ПРОГНОЗУВАННЯ ЕНЕРГОЕФЕКТИВНОСТІ В СИСТЕМІ MICROGRID

В наш час компаніями здійснюються накопичення історичних значень економічних і фізичних показників в базах даних, що істотно збільшує обсяги вхідної інформації для завдання прогнозування. Розвиток апаратних і програмних коштів надає все більш потужні обчислювальні платформи, на яких можлива реалізація складних алгоритмів прогнозування. Крім того, сучасні підходи до технічного керування пред'являють все більш жорсткі вимоги до точності прогнозування. Таким чином, завдання прогнозування ускладнюється одночасно з розвитком інформаційних технологій.

Прогнозування різних режимів роботи системи є актуальною роботою. Завдання прогнозування режиму вирішується на основі створення моделі прогнозування, який описує досліджуваний процес. На сьогоднішній день існують моделі прогнозування режимів роботи системи MicroGrid. Одні з відомих: Активно-адаптивна мережа (AAM), «AMIGO» (advanced microgrid optimization) та моделі часових рядів, які використовують класи авторегресійних і нейромережевих моделей. Істотним недоліком авторегресійного класу є велике число вільних параметрів, ідентифікація яких неоднозначна і ресурсомісткість [8]. Істотним недоліком класу нейромережевих моделей є недоступність проміжних обчислень, як наслідок, складність інтерпретації результатів моделювання. Крім того, одним з головних недоліків цього класу моделей є складність алгоритму навчання нейронної мережі [9].

Розглянемо структуру системи розподіленої генерації більш детально з точки зору виявлення особливостей і пошуку шляхів розв'язання задачі прогнозування споживання електричної енергії.

Система MicroGrid призначена для надійного забезпечення зв'язку з усіма розподіленими енергетичними ресурсами, моніторингу та керування обладнанням. MicroGrid може виконувати декілька сценаріїв, від окремих

будівель у локальній мережі до цілих міст та регіонів. Система обладнена гнучкими засобами зв'язку, які контролюють та керують всіма розподіленими енергетичними ресурсами даної системи.

Система MicroGrid об'єднує альтернативні джерела енергії (фотогенератори, вітроустановки), акумуляторні батареї, традиційні джерела енергії (дизельні генератори, паливні елементи) та керовані навантаження для забезпечення енергетичних потреб споживачів.

Завданням системи є забезпечення економії енергії, в тому числі з можливістю роботи в автономному режимі.

Для оптимізації та контролю в системі використовуються алгоритми динамічного керування, які контролюють генерацію та витрати ресурсів. Кожен розподілений енергетичний ресурс контролюється і керується локально, але його робота координується через центральну систему керування. Алгоритми керування працюють в режимі реального часу для оптимізації навантаження мережі та забезпечення надійного енергопостачання енергії [1].

### **1.1. Області застосування MicroGrid**

Основними областями застосування системи MicroGrid є регіони з необхідністю резервного постачання та регіони з несприятливим кліматом. Прикладом таких регіонів є навчальні та корпоративні містечка, стадіони, виробничі зони [3].

Одним з найбільших перших замовників рішень Microgrid стали муніципалітети штату Коннектикут. За останні два роки по території штату за погодних умов було пошкоджено централізовану мережу електропередачі.

Другою з областей застосування є віддалені регіони. Енергетичні можливості MicroGrid необхідні в самих різних місцях як в розвинених країнах, так і в країнах, що розвиваються. MicroGrid можуть гарантувати безперебійне енергопостачання банків, водоочисних споруд, аптек і державних притулків.

Якщо постачанню електроенергії загрожують природні катаклізми або проведення ліній електропередачі не можливо в даних умовах, то потрібно будувати надійніші системи.

Структура системи складається з трьох блоків електротехнічних і вимірювально-керуючих пристроїв рис 1.1.

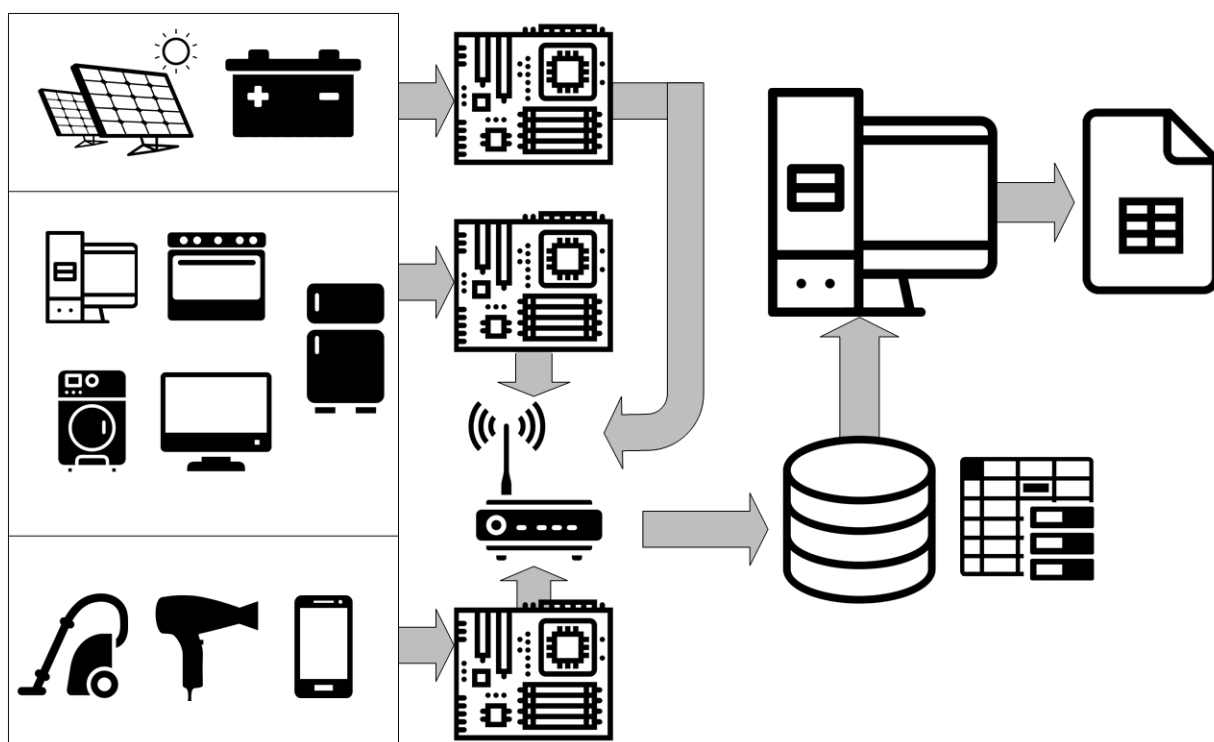


Рис 1.1 Структура системи проекту

У першому блоці знаходяться електротехнічні пристрої - генератори і споживачі електричної енергії. Ці пристрої підключені до акумуляторів та центральної електромережі. За допомогою вимірювальних пристроїв, встановлених в різних точках MicroGrid, із заданою періодичністю фіксуються і заносяться в базу даних відомості про сумарному споживанні електричної енергії MicroGrid. Серед пристроїв, що входять в перший блок, розрізняють три типи: 1) статичні пристрої, що споживають електричну енергію та підключення завжди до одного і того ж вимірювального модуля прийому даних; 2) динамічні пристрої, що споживають електричну енергію з можливістю підключення до

різних вимірювальним пристроїв, у налаштуваннях яких прописується взаємодія з пристроями динамічного типу; 3) пристрої генерації електричної енергії (УГЕЕ), статично підключені до заданих вимірювальним пристроїв які фіксують заряд акумуляторів і розряд (кількість електроенергії переданої від акумуляторів).

Другий блок містить групу вимірювальних пристроїв наступних типів:

- вимірювальні пристрої, здатні фіксувати дані про споживання електричної енергії від статичних пристроїв;
- вимірювальні пристрої, здатні фіксувати дані про споживання електричної енергії від динамічних пристроїв;
- вимірювальні пристрої, здатні фіксувати дані про генерування та накопичення електричної енергії від УГЕЕ і споживання електроенергії навантаженням.

Вимірювальні пристрої всіх типів фіксують дані про споживання і генерації електроенергії, обробляють їх (упаковують) для передачі в базу даних.

Третій блок - це блок зберігання, розрахунку та обробки даних. Цей блок містить сервер, на який приходять пакети даних, базу даних і програмне забезпечення для обробки даних.

Перший і другий блоки пов'язані між собою за допомогою електротехнічних і інформаційних комунікацій. Між другим і третім блоками здійснюється тільки інформаційна взаємодія.

Після передачі даних в базу програмне забезпечення сервера за допомогою заданого протоколу передачі «Open Data Protocol», отримує дані за деякий період (добу, місяць, квартал і т.п.) і записує їх в розрахункову матрицю регресійного аналізу. У матриці містяться наступні параметри: обсяг спожитої електроенергії, день тижня, сезон. На підставі обчислень за допомогою лінійного регресійного аналізу програма видає значення, яке відображає прогнозовану кількість залишкової електроенергії, яку можна буде продати в наступному місяці, і розрахунок грошового прибутку від такого продажу з урахуванням діючих

тарифів. Програма проводить кілька ітерацій регресійного аналізу поспіль використовуючи дані про електроспоживання з початку одного пристрою, потім іншого, суми кількох пристроїв та суми всіх пристроїв електроспоживання. Після закінчення всіх розрахунків користувачеві пропонуються кілька варіантів режимів роботи MicroGrid в розрізі окремих пристроїв або їх груп з відповідним значенням грошового прибутку, отриманої від продажу надлишку електроенергії для різних режимів. На підставі цих розрахунків користувач приймає рішення про те, який режим споживання встановити на поточний період.

## **1.2. Інформаційна система моніторингу та аналізу моделі локальної активно-адаптивної мережі**

Світові енергетичні системи розвиваються в напрямку багаторівневої автоматизації. Це дає можливість створювати основу нових електроенергетичних систем - активно-адаптивних мереж (ААС). Такі мережі відрізняються широкою функціональною насиченістю, гнучкістю методів керування джерелами енергії. ААС повністю забезпечує потреби споживачів за рахунок стійкого характеру споживання. [2]

Однак перехід великих енергетичних мережевих сегментів вітчизняної енергетики на ААС сьогодні ускладнений. Це обумовлюється високою вартістю технічних і технологічних рішень, слабким розвитком комунікаційної інфраструктури багатьох ділянок електричних мереж, відсутністю механізмів інтеграції енергетичних та інформаційних інфраструктур, слабким розвитком законодавчої бази [6, 7] та ін. Все це призводить до необхідності наукового обґрунтування режимів роботи окремих елементів системи, структури мережі в цілому в аспекті систем MicroGrid. Одним з перспективних рішень такого комплексного завдання є підхід на основі постійного системного моніторингу даних енергетичних моделей локальної ААС (MicroGrid), аналізі ключових розрахункових значень споживання та генерації енергії та встановлення

параметрів ефективного керування [10]. Крім того, статистичні дані та результати моніторингу моделі можуть використовуватися на інших етапах життєвого циклу ААС, наприклад, при пошуку оптимальних режимів роботи об'єктів електроенергетичної системи [11, 12] та виборі її конфігурацій, формуванні планів споживання енергії, обґрунтуванні рішення щодо їх зміни в облікові періоди часу.

Моніторинг енергетичних даних локальної ААС базується на основі розподілених інформаційних системах збору даних, що володіють широкими можливостями розвитку. В рамках дослідження пропонується використання методи на основі вже створеної системи - інформаційно-вимірювального комплексу JEVIs компанії Envidatec GmbH (Гамбург, Німеччина), що забезпечує функції автоматичного енергообліку та автоматизованої обробки результатів з використанням вбудованих механізмів Octave. Практика застосування інструменту JEVIs показує його високу ефективність [13, 14].

Організована система моніторингу JEGRID надає можливість отримання, передачі та обробки даних для всіх компонентів ААС. Її компонент JEGraph реалізує інструменти візуалізації, JEBench - порівняння і JECalc - аналізу даних. Структура системи моніторингу ААС на основі JEVIs, наведена на рис. 1.2.

До моделі локальної ААС підключені серверне обладнання моніторингу JEVIs Server і блок взаємодії з користувачем - Client. В рамках блоку MicroGrid показані генератори  $G_n$  і споживачі  $C_m$ , база даних - DB Master HIL MicroGrid, і FTP сервер, за допомогою якого здійснюється передача даних в форматі CSV-файлів. Зв'язок з блоком JEVIs Server проводиться по локальній мережі - LAN.

Блок JEVIs Server і Client пов'язані з допомогою технологій глобальної мережі Інтернет - WAN.

Пошук, аналіз та прийняття рішень з керування енергозбереженням в ААС виконується з метою підвищення ефективності використання енергоресурсів, а також зниження питомих витрат енергії [15]. У зв'язку з цим підвищення ефективності ААС досягається шляхом різноманітної оцінки і своєчасної



реалізації резервів мережі з урахуванням мінімізації втрат на вироблення, передачу і перетворення енергії.

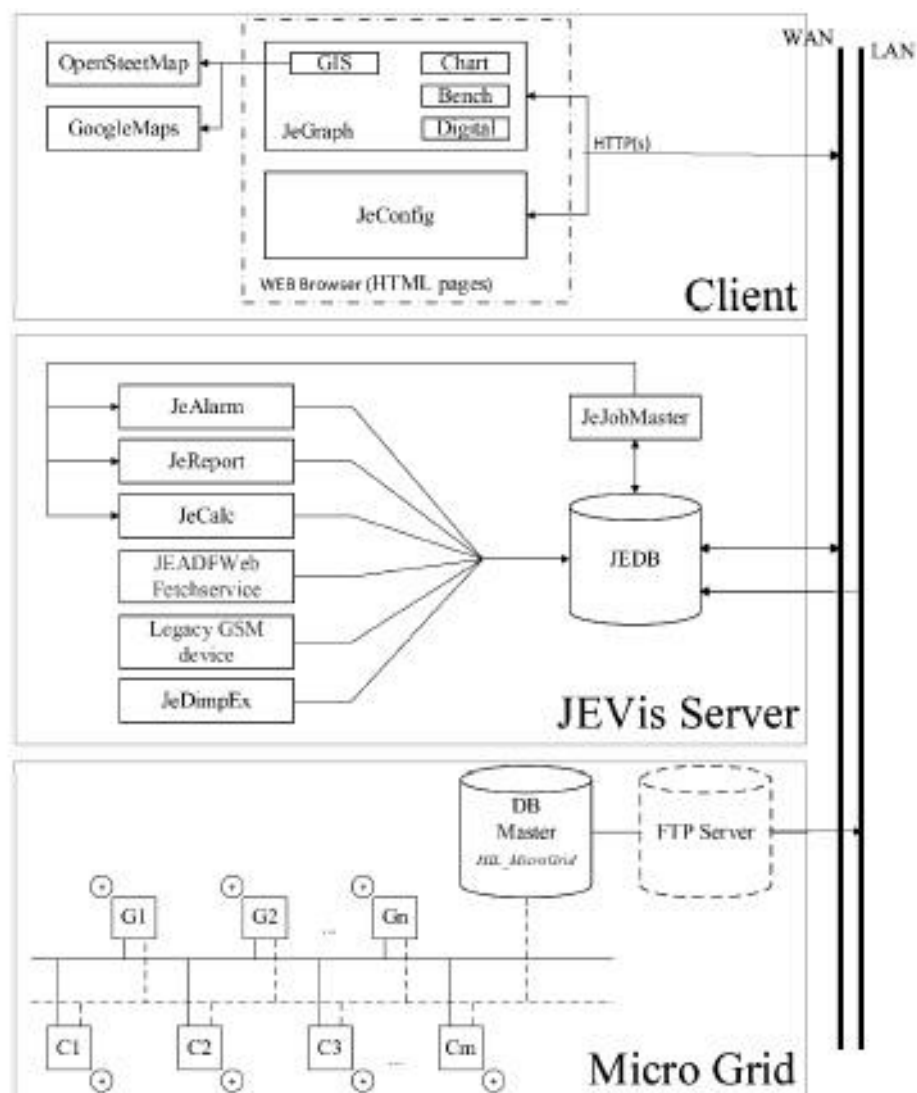


Рис 1.2 Структура системи моніторингу та аналізу енергетичних даних моделі локальної AAC

В результаті моніторингу та аналізу енергетичних даних моделі AAC за допомогою інформаційної системи на базі продукту JEVIS виявляються умови використання електроенергії як окремими споживачами, так і груп пристроїв. Дані оцінки можуть використовуватися для підвищення ефективності регулювання навантаження споживачів при розгортанні реальної мережі.

Регулювання потужності активних елементів ААС на підставі результатів аналізу енергетичних даних і застосування розвиненої мережевої інфраструктури JEVis призводить до формування гнучких локальних енергетичних систем на основі пошуку оптимального режиму роботи енергетичної інфраструктури. Незважаючи на те, що характеристики мережі постійно змінюються в часі в залежності від режимів роботи енергосистеми, система моніторингу дозволяє фіксувати зміни і кожен раз забезпечувати розрахунок цільових показників ефективності розподілу та споживання електроенергії на основі вбудованих механізмів математичних розрахунків, тим самим формуючи передумови для постійного пошуку енергоефективних конфігурацій [16].

### **1.3. Програмно-технічний комплекс оптимізації MicroGrid**

Одним з основних трендів в електроенергетиці є поступове зближення джерел генерації і споживання з утворенням енергорайона MicroGrid, з вираженими межами, з великою енергосистемою або працюють автономно. Перша причина цього процесу - розвиток малої генерації і відновлюваних джерел енергії (ВДЕ), які встановлюються поблизу споживачів і вбудовуються в локальну розподілену інфраструктуру. Другою причиною є істотне збільшення ККД при зустрічному зменшенні вартості розподілених енергоресурсів. Електричний ККД газопоршневих агрегатів в наш час досяг 40%, що вище ККД парогазових блоків 800 МВт (~ 35%), а вартість електрохімічних накопичувачів електричної енергії за останні роки зменшилася на порядок. Зрозуміло, що для ефективного функціонування Microgrid необхідно керувати всіма представленими в ній енергоресурсами.

Функції керування можуть виконувати локальні системи керування кожного пристрою, може забезпечувати спеціалістабо диспетчер енергосистеми, однак зі зростанням кількості об'єктів керування в Microgrid неможливо обійтися

без високоавтоматизованої або автоматичної інформаційно-керуючої системи. Такі системи керування повинні здійснювати координоване і оптимальне керування всіма розподіленими енергоресурсами, мінімізуючи витрати на паливо, закупівлі енергії на ринках і технічне обслуговування обладнання, підвищуючи доходи від продажу енергії та надання системних послуг (ценозалежне споживання, регулювання частоти і напруги).

Розглянемо локальну енергосистему, що складається з великої кількості об'єктів розподілених генерації, передачі, зберігання розподілу та споживання електричної енергії. Кожен момент часу необхідно виробництво такої кількості електроенергії, яке могло покрити поточне навантаження і витрати на передачу і розподіл цієї енергії. Причому у різних енергоресурсів є різна вартість виробництва енергії в залежності від типу агрегатів і палива, ККД, рівня завантаження, зносу устаткування і т. п. Одним із завдань керування такої енергосистемою є економічне або часткове керування - інакше кажучи, оптимальне керування наявним обладнанням, яке в окремому випадку зможе мінімізувати вартість електроенергії для споживачів за умови виконання ряду технологічних обмежень.

Зауважимо, що деякі параметри системи, необхідні для здійснення оптимізації режиму, такі як витратні характеристики енергоагрегатів, параметри, що характеризують знос обладнання, змінюються повільно, інші, такі як комутаційний статус обладнання - швидше, кілька разів за добу, треті, такі як потужність, що виробляється, - безліч великих кількостях разів за добу, оскільки на принципово непрогнозовані відхилення балансів потужності енергосистеми необхідно реагувати за лічені хвилини.

Завдання прогнозування вироблення електроенергії ВДЕ і споживання електроенергії є ключовим для завдань оптимізації. Завдяки прогнозуванню з'являється можливість приймати рішення на підставі поточних системних умов і на підставі умов, які складуться в MicroGrid. Таким чином, оптимізація

проводиться на інтервалі, що зв'язує минуле, теперішнє і майбутнє, - в так званому «просторі-часі».

Для прогнозування споживання на основі даних зібраних в минулому будуються і постійно уточнюються прогностичні моделі, структура яких залежить від обсягу наявних даних. Отримані прогнози незалежних параметрів оптимізації за допомогою прогностичних моделей - споживання і генерації ВДЕ - використовуються в короткостроковій і оперативній оптимізації[51].

Для вирішення завдання оперативної оптимізації використовується оперативний прогноз споживання, отриманий за допомогою методів аналізу і формування часових рядів, заснованих на авторегресійних підходах (ARIMA), або на інтерполяції найбільш нових точок траєкторії з фактичного режиму і точок з майбутнього, що формуються в завданнях більш далекого прогнозу.

Прогноз ВДЕ генерації на вітрових (BEC) і сонячних (CEC) електростанціях складається з використанням прогнозу погоди і математичних моделей, що пов'язують потужність цих електростанцій з прогнозованими метеопараметрів[54].

### **1.3.1 Довгострокова оптимізація**

Довгострокова оптимізація здійснює коригування характеристик обладнання (коригує показники в залежності від технічного стану і, головне, від напруження устаткування) і відповідно до цього впливає на пріоритети завантаження генеруючого обладнання. Наявність довгострокової оптимізації дозволяє уникнути ситуацій, коли пристрій використовувався частіше, ніж інші, і тому раніше потребують технічного обслуговування. Таким чином, алгоритм мінімізує використання не тільки неефективного обладнання, а й обладнання з найбільш високим рівнем витраченого ресурсу.

### 1.3.2 Короткострокова оптимізація

Короткострокова оптимізація вирішує завдання оптимального керування обладнанням, стратегія використання електричної енергії та покупка або продаж електроенергії на локальному ринку. Завдання даної оптимізації є більш складним, ніж проста оптимізація в момент часу. Так як, рішення про використання власного генератора для заряду акумулятора може бути не вигідним в поточний час, однак насправді це дозволить заощадити в майбутньому, коли вартість електроенергії в локальній мережі зросте. Іншим прикладом - є рішення про доцільність превентивного використання енергії накопичувачів перед режимами спаду електроспоживання або підвищеної генерації на ВДЕ з точки зору поточних режимних умов представляється недоцільним, однак неприйняття такого рішення може спричинити за собою необхідність обов'язкового відключення «безкоштовного» джерела енергії (ВДЕ) в найближчому майбутньому. Основою для вироблення стратегії короткострокової оптимізації є прогнози споживання і некерованої генерації.

Приклад функції, яку необхідно мінімізувати при наявності  $m$  генераторів у внутрішній мережі MicroGrid і зовнішньої мережі, записується таким чином:

$$C_{\Sigma} = \sum_{j=1}^m \left( \sum_{i=1}^{96} C_{\Gamma_{(i)}} + C_{\text{пуск}\Sigma} + C_{\text{стоп}\Sigma} \right) + \sum_{j=0}^k \sum_{i=1}^{96} C_{\text{сеть}_i} \rightarrow \min ,$$

де перший доданок відповідає за вартість роботи  $C_{\Gamma_{(i)}}$ , пусків  $C_{\text{пуск}\Sigma}$  та зупинок  $C_{\text{стоп}\Sigma}$  генераторів, а другий - за вартість покупки електроенергії з зовнішньої мережі.

Мінімізація цільової функції повинна здійснюватися при обов'язковому дотриманні обмежень у формі рівностей та нерівностей.

Типовим обмеженням в формі рівностей є умова балансу активної потужності в кожен момент часу  $i$ , що визначається рівнянням:

$$\sum_{j=1}^m P_{\Gamma_j} \pm P_{\text{мереж}} \pm \sum_{j=0}^k P_{\text{МНЕ}} - P_{\text{наван.}\Sigma} - P_{\text{втрати}} = 0,$$

де  $P_{Gj}$  - поточна потужність кожного генератора,  $P_{мереж}$  - потужність, що купується або продається в мережу,  $P_{МНЕ}$  - потужність, яку віддає або споживає мережевий накопичувач. Дане рівняння використовується при вирішенні завдання короткострокової оптимізації, за умові лінійному уявленні енергосистеми. У загальному випадку облік балансів потужності для нелінійного або лінеаризованого уявлення електричної мережі здійснюється за рахунок вирішення систем рівнянь усталеного режиму, що забезпечують зведення балансів в кожному вузлі мережі по активній та реактивній потужності.

Обмеження в формі нерівностей дозволяють отримати індивідуальні обмеження устаткування: мінімальну і максимальну потужність, обмеження за часом, максимальний заряд, , пов'язаних з особливостями пусків і зупинок устаткування, мережеві обмеження, інтегральні обмеження на накопичену енергію, запаси палива і допустиму швидкість зміни режиму генерації або керованої навантаження.

Змінні для необхідності розрахунку режимів виникають через те, що генератор може перебувати в кінцевій кількості станів (включений, вимкнений, в режимі гарячого резерву), причому в деяких з них він видає потужність, а в деяких - ні. Так як режим роботи включення або вимикання генератора не проводиться миттєво, воно впливає на деякі накладні витрати, та, відповідно, рішення про зміну його стану, прийняте зараз, впливає на майбутній період.

Можуть бути різні підходи до вирішення такого завдання. Зокрема, добре себе показав метод динамічного програмування (DP) [25]. Відповідно до нього спочатку завдання вирішується в зворотному часі, визначаючи для кожної точки фазового простору (в даному випадку в просторі стану параметрів генераторів, кількості енергії в накопичувачах і обсягу закупівель з зовнішньої мережі) найменшу вартість переходу в кінцевий стан системи  $V(s, t)$ . Для кінцевого моменту часу вартість переходу завжди дорівнює нулю, для передостаннього є можливість її визначити, знаючи значення для останнього, і т. п. Знаючи значення функції ціни в перший момент часу, можна вибрати оптимальний

початковий стан в перший момент, другий і так до кінця. Метод дозволяє гарантовано знайти оптимальне рішення в межах заданої точності. Нелінійність задачі не є обмежуючим фактором, час розрахунку завжди лінійно залежить від періоду розрахунку. Одним з головних недоліків є те, що з ростом числа незалежно керованих накопичувачів час розрахунку стає незадовільно довгим.

Тому в даному випадку використовується класичний метод вирішення задачі MILP (Method for Solving the Integer Problem) - метод гілок і меж. При цьому функція лінеаризується, криві вартості замінюються кусково-лінійними функціями. Загальний принцип методу гілок і меж наступний: по черзі фіксуються деякі набори цілочисельних змінних, інші цілочисельні змінні вважаються неперервними і використовуються при розв'язанні задачі лінійного програмування. Якщо цільова функція вийшла менше, ніж у якогось отриманого раніше рішення, то відкидається ціла «гілка» рішень, що містить даний фіксований набір цілочисельних змінних. Результатом роботи короткострокової оптимізації є графік роботи обладнання на 24 години вперед.

Спираючись на вищевикладені методики, був розглянутий модуль економічної оптимізації та третинного керування для інформаційно-керуючої системи нового покоління AMIGO (Advanced Microgrid Optimization), які дозволяють забезпечити всі вищевказані можливості оптимального керування в рамках локальних енергосистем та отримати суттєвий економічний ефект - зниження витрат на енергопостачання на 5-20%.

#### **1.4. Фактографічні, експертні і комбіновані методи прогнозування**

Метод прогнозування являє собою послідовність дій, які потрібно зробити для отримання моделі прогнозування. Залежно від наявності інформаційних даних методи прогнозування поділяють на фактографічні, експертні і комбіновані.

Фактографічні (формалізовані) методи ґрунтуються на достатньому інформаційному матеріалі про об'єкт прогнозування та його минулий розвиток. Формалізовані методи засновані на математичних моделях і поділяються моделі предметної області, як-то механіки, термодинаміки, електротехніки, тощо та моделі часових рядів, які шукають залежності всередині самого процесу. Для моделей предметної області властивий індивідуальний підхід в розробці. Моделі часових рядів є універсальними для різних предметних областей [17].

Експертні методи застосовують у тих випадках, коли інформаційний матеріал, який характеризує розвиток об'єкта у минулому, недостатній. Вони побудовані на інформації, отриманій за оцінками спеціалістів-експертів [20].

Комбіновані методи прогнозування об'єднують експертні і фактографічні методи. Прикладом таких методів може бути метод *Pattern matching*, при якому експерти формулюють колективні судження на основі використання принципу “дерева цілей”. Інколи до цієї групи відносять моделі на базі нечіткої логіки. Нечітка логіка є розширенням експертних систем, а тому вона потребує перекладення досвіду в нечіткі правила.

Серед розглянутих методів для прогнозування енергоспоживання в системах *MicroGrid* найбільш придатним є формалізований метод, оскільки при цьому можливо створити автоматизовано систему прогнозу. Крім того, поведінка кривої енергоспоживання описується функцією часу, тому обираємо групу моделей часових рядів. Вибір конкретної моделі потребує подальшого їх дослідження і порівняння по точності прогнозу в конкретних задачах прогнозування електроспоживання.

Моделі часових рядів можна розділити на дві групи: статистичні та структурні [17]. Статистичні метод – це метод після проведення аналізу якого на даних які були отримані в минулому може бути отримано рівняння, що відображає залежність між енергоспоживанням і пов'язаними зовнішніми факторами. Як правило, статистичні методи можуть передбачити енергоспоживання звичайного дня дуже добре, але в них не закладено достатньо



гнучкості, щоб аналізувати святкові та інші нерегулярні дні, що стає одним з головних недоліків цього методу. В структурних моделях залежність майбутнього значення від минулого задається в вигляді деякої структури та правил переходу.

### **1.5. Моделі часових рядів**

Група статистичних моделей в які входять економетричні, регресивні та авторегресивні моделі.

Економетричні моделі засновані на згладжуванні, експонентному згладжуванні [19]. Серед них розрізняють наступні: «наївна» модель прогнозу; модель обчислення прогнозу на основі середнього згладжування; модель обчислення прогнозу на основі експонентного згладжування.

Моделі цього типу роблять прогнози на основі лінійних комбінацій споживання значення від «подібних» днів [18].

При створенні "наївних" моделей передбачається, що деякий період прогнозованого часового проміжку краще всього описує майбутнє. Модель на основі згладжування визначає залежність прогнозу від усіх розглянутих даних, причому вплив даних на прогноз експоненціально зменшується із "віком" даних. Розвитком моделей цього типу є моделі двохпараметричного згладжування Хольта і Брауна, а також трьохпараметричної моделі Вінера.

Даний клас моделей частіше інших використовується для довгострокового прогнозування [20].

Регресивні моделі поєднують в собі кілька незалежних функцій, утворюють лінійну функцію, що допомагає інтерпретувати залежності між різними факторами. Для прогнозування електроспоживання від альтернативних джерел живлення, таких як фото та вітрогенератори будується модель дерева з використанням даних про погоду і графік даних для прогнозування споживання енергії. Він оцінює вплив різних комбінацій ознак на точність прогнозування. Зазначимо, що при використанні лінійних регресивних моделей результат

прогнозування може бути отриманий швидше, ніж при використанні інших моделей. Нелінійні регресивні моделі характеризуються великою трудомісткістю визначення параметрів моделі [21].

Автогресивні моделі (Бокса-Дженкінса) є найбільш популярними з статистичних моделей [22]. У відкритому доступі легко знайти приклади застосування авторегресивних моделей (ARIMA, ARIMAX, GARCH, ARDLN) для вирішення задач прогнозування за допомогою часових рядів різних предметних областей [23]. Моделі спираються тільки на інформацію, яка міститься в минулих даних прогнозованих рядів, що обмежує можливості алгоритму. В методології ARIMA не передбачається якої-небудь чіткої моделі для прогнозування даної часової серії. Задається лише загальний клас моделей, що дозволяють виражати поточне значення змінної через її попередні значення. Наступним кроком алгоритм, вибирає найбільш відповідну модель прогнозування.

Група структурних моделей включає в себе моделі на базі ланцюгів Маркова та на базі класифікаційних регресійних дерев. Нечітка логіка є розширенням експертних систем, а тому вона як потребує перекладенні досвіду в нечіткі правила.

Прогнозування на базі штучних нейронних мереж (Artificial Neural Networks (ANNs)) прагне визначити залежності між вхідним набором даних і вихідних. Вони добре підходять при в роботі з нелінійними залежностями між енергоспоживанням і зовнішніми факторами, але їх недолік полягає в «перетренованості мережі» (overfitting) і великому часу навчання.

Моделі прогнозування на основі ланцюгів Маркова припускають, що майбутній стан процесу залежить тільки від його поточного стану і не залежить від попередніх. Моделі використовують матрицю ймовірностей переходів системи з одного стану в інший. Структура ланцюга Маркова та ймовірності переходу станів визначають залежність між майбутнім значенням та його поточним значенням.

Моделі на базі класифікаційно-регресивних дерев розроблені для моделювання процесів, на які впливають як безперервні зовнішні фактори, так і категоріальні [25].

Крім наведеної класифікації основних моделей для прогнозування часових рядів, можуть застосовуватися наступна класифікація фактографічних моделей, яка поділяє їх на дві великі групи [26]:

1. Моделі однофакторного прогнозування. Вони об'єднують дві великі підгрупи, а саме: моделі прогнозування тренду (експоненціональне згладжування, узагальнене експоненціональне згладжування, гармонічні ваги, спектральні методи, метод ковзної середньої, метод Бокса-Дженкінса, метод скінченних різниць, різницевих рівнянь, оптимальних фільтрів, сплайн-функції) і методи прогнозування випадкової компоненти (метод авторегресії, ймовірнісний метод, метод ланцюгів Маркова).

2. Моделі багатофакторного прогнозування, до яких належать регресивні моделі (адаптивні регресії, регресивні K-моделі), адаптивне згладжування, факторний аналіз, метод групового врахування аргументів, імітаційні моделі, багатовимірні фільтрація.

### **1.6. Переваги і недоліки основних моделей прогнозування**

Переваги і недоліки основних моделей систематизовані в табл. 1.1.

Аналіз досліджень щодо прогнозування споживання енергії в системах SmartGrid та MicroGrid показав, що в них використовувалися прості моделі усереднення; статистичні моделі (наприклад, регресія і часові ряди) і моделі штучних нейронних мереж. В роботах [27, 28] показано переваги комбінованої моделі, яка містить два етапи - фільтрації нестационарної складової енергоспоживання та нейронної мережі з пам'яттю.

Таблиця 1.1.

## Порівняння моделей прогнозування

Моделі	Переваги	Недоліки
<b>Статистичні моделі</b>		
Регресивні	Простота, гнучкість, прозорість моделювання; одноманітність аналізу і проектування	Складність визначення функціональної залежності; трудомісткість знаходження коефіцієнтів залежності; відсутність можливості
Авторегресійні	Простота, прозорість моделювання; одноманітність аналізу і проектування; безліч прикладів застосування	Велика трудомісткість і ресурсомісткість ідентифікації; неможливість моделювання нелінійностей; низька адаптивність
Моделі експоненційного згладжування	Простота моделювання; одноманітність аналізу і проектування	Недостатня гнучкість; вузька застосовність моделей
<b>Структурні моделі</b>		
Нейромережеві моделі	Нелінійність моделей; масштабованість, висока адаптивність; одноманітність аналізу і проектування; безліч прикладів застосування	Відсутність прозорості; складність вибору архітектури; жорсткі вимоги до навчальної вибірки; складність вибору алгоритму навчання; ресурсомісткість процесу навчання
Моделі на базі ланцюгів Маркова	Простота моделювання; одноманітність аналізу і проектування	Неможливість моделювання процесів з довгою пам'яттю; вузька застосовність моделей
Моделі на базі класифікаційних регресивних дерев	Масштабованість; швидкість і простота процесу навчання; можливість враховувати категоріальні змінні	Неоднозначність алгоритму побудови дерева; складність питання зупинки

Фільтрації нестационарної складової за допомогою вейвлет-перетворення дозволило зменшити кількість входів нейронної мережі при забезпеченні заданого коефіцієнта схожості вихідної функції енергоспоживання та відфільтрованої.

### Висновки до першого розділу

Для того щоб оптимізувати енергоспоживання в мережі MicroGrid, його необхідно зпрогнозувати на деякий період вперед. Прогнозування енергоспоживання має особливу вагу при автономній роботі, оскільки

відновлювальні джерела живлення характеризуються високою нестабільністю рівня генерованої потужності, що залежить від кліматичних умов. Тому актуальною задачею для керування навантаженнями MicroGrid є прогнозування енергоспоживання при різних умовах зовнішнього середовища.

В даний час завдання прогнозування залишку електричної енергії в Micro Grid можна вирішувати багатьма методами. Серед розглянутих методів для прогнозування енергоспоживання в системах MicroGrid найбільш придатним є метод регресійного аналізу, оскільки при цьому можливо створити автоматизовано систему прогнозу. Крім того поведінка кривої енергоспоживання описується функцією часу, тому обираємо групу моделей часових проміжків. Вибір конкретної моделі потребує подальшого їх дослідження і порівняння по точності прогнозу в конкретних задачах прогнозування електроспоживання.

## РОЗДІЛ 2. РЕГРЕСІЙНИЙ АНАЛІЗ

Серед розглянутих методів для прогнозування енергоспоживання в системах MicroGrid найбільш придатним є метод кореляційно-регресійного аналізу, оскільки при цьому можливо створити автоматизовану систему прогнозу. Поведінка кривої енергоспоживання описується функцією часу, тому обираємо групу моделей часових проміжків. Вибір данної моделі потребує подальшого їх дослідження і порівняння по точності прогнозу в конкретних задачах прогнозування електроспоживання.

### 2.1. Основні поняття

Регресійний аналіз - розділ математичної статистики, присвячений методам аналізу залежності однієї величини від іншої. На відміну від кореляційного аналізу, регресійний аналіз не дозволяє виявити існування істотного зв'язку між сигналами, а здійснює пошук моделі цього зв'язку, вираженої у функції регресії.

Регресійний аналіз використовується в тому випадку, якщо відношення між змінними можуть бути виражені кількісно у виді деякої комбінації цих змінних. Отримана комбінація використовується для передбачення значення, що може приймати цільова (залежна) змінна, яка обчислюється на заданому наборі значень вхідних (незалежних) змінних. У найпростішому випадку для цього використовуються стандартні статистичні методи, такі як лінійна регресія. Більшість реальних моделей не вкладаються в рамки лінійної регресії. Розміри продажів чи фондові ціни дуже складні для передбачення, оскільки можуть залежати від комплексу взаємозв'язків множин змінних. Таким чином, необхідні комплексні методи для передбачення майбутніх значень.

Мета регресійного аналізу - встановити конкретну аналітичну залежність одного або декількох результативних показників від одного або декількох ознак-

факторів. Отримане при цьому рівняння регресії використовується для змістовного опису досліджуваного процесу, прогнозування, вибору оптимального варіанту і т.д. Якщо в рівняння регресії включені ознаки-фактори, що враховують і можливе випадкове поводження результативної ознаки, то такий вираз являє регресійну модель явища чи процесу. Найбільше застосування отримали рівняння регресії, що відображають взаємозв'язок одного результативного ознаки з однією (парна регресія) або декількома (множинна регресія) ознаками-факторами.

Після попередньої обробки статичних даних виконується формування рівняння регресії, що пов'язує вихідну змінну  $Y$  та кожний з аргументів  $X$ . Загальна схема послідовних дій полягає в наступному:

- за дослідженими даними складається кореляційна таблиця і кореляційне поле в декартовій системі координат;
- обчислюються середньоінтервальні значення  $\bar{Y}_{x_i}$  і будується емпірична лінія регресії  $\bar{Y}_{cp}$  по змінній  $X_i$ , що характеризує зміну середнього значення вихідної функції під впливом  $i$ -го чинника.

Середньоінтервальні значення  $\bar{Y}_{cp}$  визначаються за допомогою співвідношення

$$\bar{Y} = \frac{\sum m_i y_i}{\sum m_i}. \quad (2.1)$$

Доцільно використовувати спрощений метод обчислень шляхом переходу до нових значень функції

$$\bar{Y}'_i = \frac{Y_i - C_{y_i}}{\Delta Y}, \quad (2.2)$$

де  $C_{y_i}$  – вибраний центр переходу до нових значень. Перетворені змінні переводяться в натуральні:

$$\bar{Y}_{cp} = \bar{Y}'_{cp} \cdot \Delta Y + C_{y_i}, \quad (2.3)$$

де  $\Delta Y$  – величина інтервалу.

Середньоінтервальні значення функції  $\bar{Y}_{cp}$  відображаються на кореляційному полі у вигляді інтервалів змінної  $X_i$ . Ці точки з'єднуються, і отримана ламана лінія має назву емпіричної. За законом великих чисел можна стверджувати, що емпірична лінія регресії все більше згладжуватиметься при зростанні числа спостережень.

Граничне положення емпіричної лінії регресії, до якого вона прямує при необмеженому збільшенні числа спостережень, має назву теоретичної лінії регресії, а процес її знаходження – вирівнюванням емпіричної лінії регресії.

Важливу роль в кореляційному аналізі відіграє підбір форми математичного рівняння, що найкращим чином описує досліджуваний процес.

Математичні функції  $\bar{Y} = f(x_i)$  для опису залежності можуть бути різноманітними. Найширше розповсюдження знайшли лінійні рівняння регресії вигляду:

$$\bar{Y} = kx_i \pm b. \quad (2.4)$$

Параметри лінійного рівняння легко визначаються, рівняння зручне в застосуванні, теорія лінійної кореляції вивчена найбільш докладно. Однак лінійна залежність має свої специфічні властивості, обмежують сферу застосування лінійного регресійного аналізу при вивченні та врахуванні економічних явищ.

Основні властивості лінійної функції:

- швидкість зміни  $Y$  із зростанням  $X$  постійна, не залежить від величини  $X$ , тобто  $\frac{\Delta y}{\Delta x} = const$ ;
- абсолютні максимум і мінімум  $Y$  в будь-якому діапазоні зміни  $X$  досягаються на його кінцях.

З цього випливає, що різноманітні процеси, швидкість перебігу яких змінна і залежить від досягнутого рівня, тільки приблизно можуть описуватися за допомогою лінійної функції, якщо абсолютний максимум або мінімум величини, яка вивчається, досягається всередині даного діапазону зміни  $X$ , то



за допомогою лінійної функції він виявлений бути не може. Ця остання обставина особливо суттєва при вирішенні задач оптимізації.

Далі лінійна регресія  $Y$  по  $X$  припускає виконання наступних умов:

- дисперсія величини  $Y$ , відповідна даному значенню  $X$ , постійна, тобто  $\sigma_{yx}^2 = \sigma_x^2$ , або пропорційна даній функції від  $X$ , тобто  $\sigma_{yx}^2 = \sigma_x^2 \cdot h(x)$ .

Вказані умови на практиці виконуються рідко. Отже, складність реальних процесів, нерівномірність їх протікання, наявність екстремальних значень і різнонаправленість дії окремих взаємозв'язаних чинників у більшості випадків не дозволяють достатньо обґрунтовано застосовувати лінійні методи. Це призводить до необхідності розрахунку нелінійних рівнянь.

Виходячи з фізичної, технологічної, економічної сутності дослідженого процесу і враховуючи вид лінії регресії, робиться гіпотеза про залежність функціональної ознаки  $Y$  від аргументу  $X$ , забезпечуючи найкраще наближення розрахункових результатів до початкових даних.

Доцільно використовувати дво- і трипараметричними рівняннями. Для моделювання показників, що монотонно зростають, можна отримати наближення двопараметричними значеннями, а моделі трипараметричні при коректному їх підборі виконуються для досить точного опису більшості процесів, що допускають єдиний екстремум. Монотонність або екстремальність залежності повинна бути обумовлена характеристиками процесу і підтверджуватися виглядом лінії регресії.

Використовувати багатопараметричні рівняння не є доцільним, хоча за допомогою максимуму функції можна отримати близьке наближення до початкових даних. Проте таким чином описується не стільки велика тенденція, скільки випадкові відхилення. Такі функції можуть мати максимуми і мінімуми, не виправдані по суті. Крім того, як складання таких функцій, так і їх застосування для практичних розрахунків різко ускладнюється.

Більшість математичних функцій, що використовуються для опису техніко-економічних показників, шляхом функціональних перетворень  $Y$  по  $X$

(роздільно або одночасно) можуть бути зведені до лінійного вигляду. При цьому метод перетворень залежить від форми зв'язку.

Гіпербола вигляду  $\bar{y} = \frac{k}{x} + b$  перетвориться в лінійну шляхом заміни.

Статична функція вигляду  $\bar{y} = b \cdot k^x$  перетвориться в лінійну шляхом логарифмування. У результаті маємо  $\lg \bar{y} = \lg b + k \lg x$ , рис 2.1. Позначимо  $\bar{y}' + \lg \bar{y}, b' = \lg b, x' = \lg x$ . У результаті маємо  $\bar{y}' = kx' + b'$ , рис 2.2.

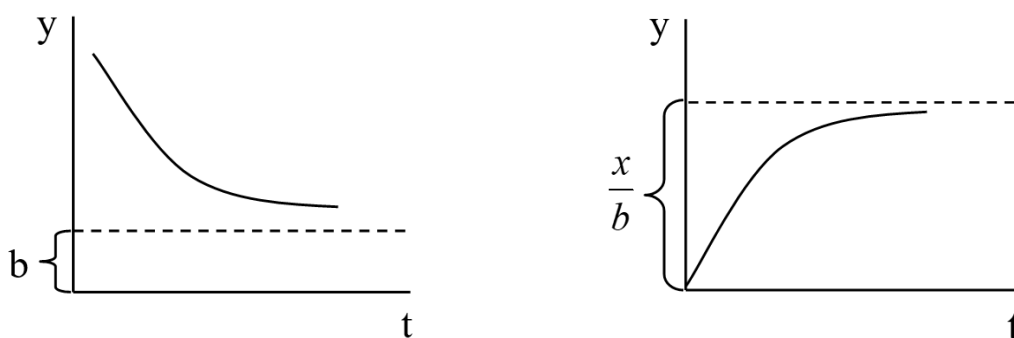
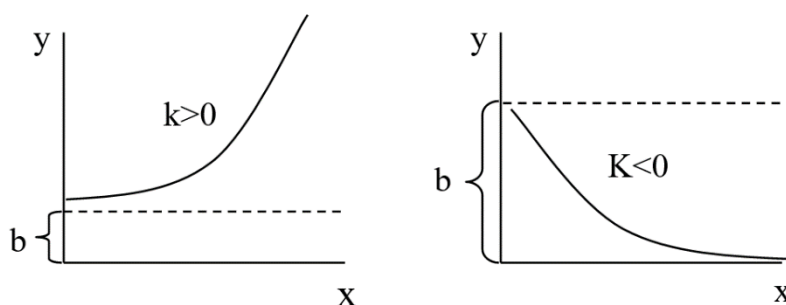


Рис. 2.1



ж

Рис. 2.2.

Показова функція виду  $y = b \cdot e^{k \cdot x}$  перетвориться в лінійну логарифмуванням  $\lg \bar{y} = \lg b + k \lg e$ , рис 2.3.

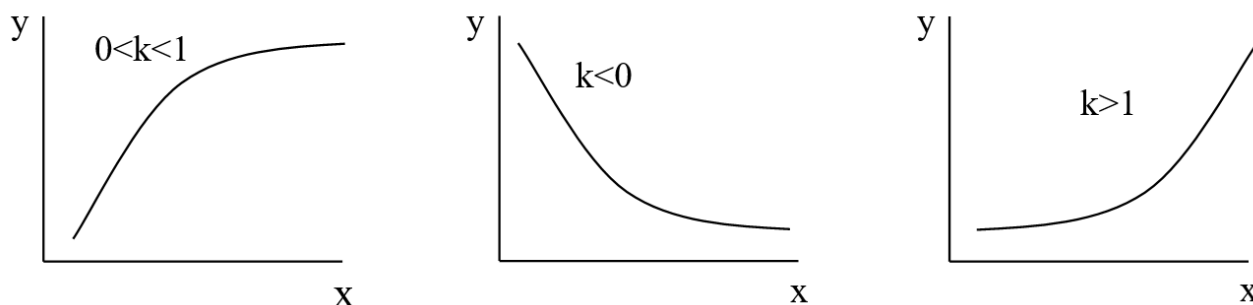


Рис. 2.3.

Позначимо  $\bar{y}' + \lg \bar{y}, b' = \lg b, k' = k \lg e$ , при цьому  $\lg e = 0.4343$ . У результаті маємо  $y_1 x + b_1$ .

Теоретична лінія регресії може бути подана у вигляді плавної кривої, яка кількісно виражає зв'язок між середніми інтервальними значеннями  $\bar{y}_{cp}$  і відповідними значеннями  $X$  (аргументами). Процес знаходження невідомих параметрів теоретичної залежності є однією з важливих проблем теорії кореляції і регресії.

Існує декілька способів встановлення аналітичної залежності, точніше, знаходження невідомих параметрів рівняння: графічний, методи середніх, найменших квадратів та інші.

При графічному методі залежність визначають за допомогою лінійки так, щоб по обох сторонах проведеної лінії розташовувалося приблизно однакова кількість точок. Якщо характер розташування точок вказує на наявність криволінійної залежності, то використовується лекало, або шляхом функціональних перетворень змінних нелінійна залежність зводиться до лінійного вигляду.

Числове значення коефіцієнта залежності функції  $\bar{y} = kx + b$  визначається за формулою

$$k = \frac{y - b}{x}, \quad (2.5)$$

де  $y$  і  $x$  - похідні числові значення за графіком. Графік на рис 2.4.

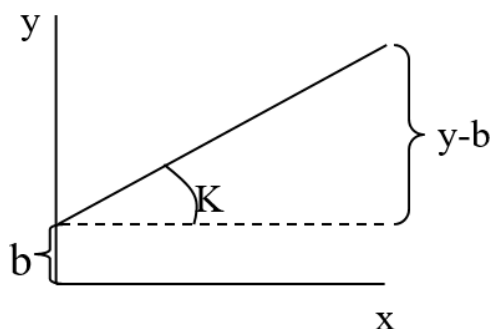


Рис. 2.4.

Відшукавши дві точки, по них проводять пряму.

Якщо лінійну пряму розташувати так, щоб точки однаково розташовувалися по обидві сторони від неї, то параметри лінійного рівняння можна визначити з рівняння

$$\frac{y - y_1}{y_2 - y_3} = \frac{x - x_1}{x_2 - x_3}, \quad (2.6)$$

де  $y_1, x_1, y_2, x_2$  – значення координат двох точок, які найкращим чином "вписуються" у пряму.

Сутність середньоарифметичного способу полягає в знаходженні середньоарифметичного значення від всіх  $n$  значень  $y$  і  $x$ , а також додаткових середніх для точок, чисельні значення яких менше  $x_{cp}^b$  і  $y_{cp}^b$  і більше  $x_{cp}^H$  та  $y_{cp}^H$ . У цьому випадку сума відхилень відстаней точок по обидві сторони від теоретичної лінії, визначених на осі ординат, дорівнює нулю.

Кутовий коефіцієнт теоретичної лінії знаходять за формулою

$$k = \frac{y_{cp}^b - y_{cp}^H}{x_{cp}^b - x_{cp}^H}, \quad (2.7)$$

де  $x_{cp}^b$  і  $y_{cp}^b$  – верхня середня (координати точок);  $x_{cp}^H$  та  $y_{cp}^H$  – нижня середня.

Значення вільного члена  $b$  знаходять за графіком.

Основними недоліками графічного методу визначення математичної залежності слід вважати суб'єктивність «найкращого розташування» лінійної прямої. Якщо на тривалість виконання роботи впливають одночасно кілька чинників, то графічний метод можна застосовувати тоді, коли є можливість заздалегідь визначити залежність роздільно по кожному чиннику за умови, що значення решти чинників залишаються постійними.

Для обчислення параметрів рівняння виду  $\bar{y} = kx + b$  частіш за все користуються методом найменших квадратів. При цьому ставиться умова, щоб сума квадратів відхилень (відстаней) всіх досліджених точок від ординат, обчислених за рівнянням прямої  $\varepsilon_i$ , була мінімальною. Іншими словами, пряма

повинна проходити якомога ближче до вершин емпіричної лінії регресії. Це означає, що параметри  $K$  та  $b$  керування регресії треба визначити з рівняння:

$$\varepsilon_i = \sum_{i=1}^n (y_i - \tilde{y}_i) = \min, \quad (2.8)$$

де  $y_i$  – ординати досліджуваних точок;  $\tilde{y}_i$  – ординати розрахункових точок, визначені за рівнянням регресії  $\bar{y} = kx_i + b$  таким чином.

$$\varepsilon_i = \sum_{i=1}^n [y_i - (kx_i + b)]^2 = F(k, b) \min. \quad (2.9)$$

Умовою екстремуму даної функції слід вважати рівність нулю часткових виробничих, узятих за параметрами  $K$  та  $b$ :

$$\begin{aligned} \frac{dF}{dk} = 0 \text{ та } \frac{dF}{db} = 0, \quad [F(u)] = F_u(u) \cdot u' \\ \text{звідси} \quad \frac{dF}{dk} = -2 \sum_{i=1}^n [y_i - (kx_i + b)] x_i = 0 \\ \frac{dF}{db} = -2 \sum_{i=1}^n [y_i - (kx_i + b)] = 0 \end{aligned} \quad (2.10)$$

Скоротивши на (-2) і розкривши квадратні дужки, отримаємо систему лінійних рівнянь

$$\begin{aligned} \sum_{i=1}^n y_i x_i &= k \sum_{i=1}^n x_i^2 + b \sum_{i=1}^n x_i \\ \sum_{i=1}^n y_i x_i &= k \sum_{i=1}^n x_i + bn \end{aligned}, \quad (2.11)$$

підставивши сюди чисельні значення відповідних величин, знайдемо параметри  $K$  та  $b$ .

У разі лінійної залежності геометричне і алгебраїчне значення коефіцієнта регресії полягає в тому, що він кількісно характеризує на скільки в середньому змінюється  $y$  при зміні  $x_i$  на одиницю свого вимірювання. Чим більше чисельні значення коефіцієнта регресії, тим більше відносний приріст функції при зміні аргументу.

При знаходженні параметрів параболи виду  $\bar{y} = ax^2 + bx + c$  необхідно скласти і вирішувати систему з трьох нормальних рівнянь, яке розв'язується, виходячи з вимоги методу найменших квадратів, тобто  $\varepsilon_i = \sum_{i=1}^n (y_i - \tilde{y}_i)^2 = \min$ .

Підставляючи  $\bar{y} = ax^2 + bx + c$ , маємо

$$\varepsilon_i = \sum_{i=1}^n \left[ y_i - (ax^2 + bx + c) \right]^2 = \min = F(a, b, c). \quad (2.12)$$

Знаходимо часткові похідні  $\frac{dF}{da}, \frac{dF}{db}, \frac{dF}{dc}$  і прирівнюємо їх до нуля

$$\begin{aligned} \frac{dF}{da} &= -2 \sum_{i=1}^n \left[ y_i - (ax^2 + bx + c) \right] x_i^2 = 0 \\ \frac{dF}{db} &= -2 \sum_{i=1}^n \left[ y_i - (ax^2 + bx + c) \right] x_i = 0. \\ \frac{dF}{dc} &= -2 \sum_{i=1}^n \left[ y_i - (ax^2 + bx + c) \right] = 0 \end{aligned} \quad (2.13)$$

Після відповідного перетворення маємо

$$\begin{aligned} \sum_{i=1}^n y_i x_i^2 &= a \sum_{i=1}^n x_i^4 + b \sum_{i=1}^n x_i^3 + c \sum_{i=1}^n x_i^2 \\ \sum_{i=1}^n y_i x_i &= a \sum_{i=1}^n x_i^3 + b \sum_{i=1}^n x_i^2 + c \sum_{i=1}^n x_i. \\ \sum_{i=1}^n y_i &= a \sum_{i=1}^n x_i^2 + b \sum_{i=1}^n x_i + cn \end{aligned} \quad (2.14)$$

Не важко помітити, за яким правилом складається система нормальних рівнянь для знаходження невідомих параметрів шуканої функції.

## 2.2. Оцінка тісноти, суттєвості й лінійності (нелінійності) зв'язку між змінними

Тіснота зв'язку між змінними характеризується ступенем відхилення (розсіяння) досліджуваних точок біля теоретичної лінії регресії. Чим ближче окремі спостереження розташовані до теоретичної лінії регресії, тим більше повна залежність  $y$  по  $x$ .

Кутовий коефіцієнт лінійного кореляційного зв'язку між  $y$  та  $x$ , який показує, на скільки одиниць в середньому зміниться функція, якщо аргумент збільшується (зменшується) на одиницю свого вимірювання не може служити показником тісноти зв'язку між змінними. У цьому випадку його чисельне значення залежить від прийнятих одиниць вимірювання змінних.

Для отримання оцінки тісноти зв'язку між змінними використовується емпіричне кореляційне відношення ( $\eta_{y/x}^2$ ), яке є часткою дисперсії (коливаємості) функції  $y$  за рахунок впливу даного аргументу  $x$ . У даному випадку загальна (повна) дисперсія розкладається на дві частини – дисперсію усередині кожного інтервалу зміни функції  $\sigma_{y/x}^2$ , яка не залежить від впливу  $x$ , і дисперсію середніх значень функції  $\delta \frac{2}{y}$ , яка викликана впливом аргументу, тобто

$$\sigma_y^2 = \sigma_{y/x}^2 + \delta \frac{2}{y}. \quad (2.15)$$

Звідси формула для оцінки тісноти зв'язку між змінними має вигляд

$$\eta_{y/x}^2 = \frac{\sigma_{\tilde{y}}^2}{\sigma_y^2} = \frac{\frac{1}{n} \sum_{i=1}^n (\tilde{y}_i - \tilde{y})^2}{\sigma_y^2}, \quad (2.16)$$

а в разі згрупованих даних

$$\eta_{y/x}^2 = \frac{\frac{1}{n} \sum_{i=1}^n (\tilde{y}_i - \tilde{y})^2 m_i}{\sigma_y^2}, \quad (2.17)$$

де  $\tilde{y}_i$  - розрахункове значення функції;  $\tilde{y}$  - середнє значення функції за вибіркою;  $n$  - обсяг вибірки;  $k$  - кількість інтервалів зміни функції;  $m_i$  - число спостережень  $y$  в кожному інтервалі зміни.

Кореляційне відношення не залежить від одиниць вимірювання змінних, що вивчаються. Воно показує, яку частину загальної дисперсії  $\sigma_y^2$  можна віднести за рахунок зміни аргументу на одну  $\sigma_x^2$ .

При цьому характеристика  $\eta_{y/x}^2$  тим точніше визначає частку впливу  $x$  на загальну дисперсію  $y$ , чим менше варіюється залишкова дисперсія  $\sigma_{y/x}^2$  при кожному  $X$ . Якщо  $\eta_{y/x} = 1$ , то має місце функціональна залежність  $y$  від  $x$ . Якщо  $\eta_{y/x} = 0$  -  $y$  кореляційно не залежить від  $x$ .

У разі лінійної залежності змінних дисперсію середніх значень функції можна записати у вигляді

$$\sigma_y^2 = \frac{\sum_{j=1}^k (\tilde{y}_j - \bar{y})^2 m_j}{n} = \frac{\sum_{i=1}^k (kx_i + b - k\bar{x} - b)^2 m_i}{n} = \frac{\sum_{j=1}^k (x_j - \bar{x})^2 mn}{\sum_{i=1}^k m_i}. \quad (2.18)$$

Показник тісноти зв'язку для випадку лінійної залежності буде

$$\eta_{y/n}^2 = \frac{\sigma_y^2}{\sigma_y^2} = \frac{k^2 \sigma_x^2}{\sigma_y^2}, \text{ а } \eta_{y/x} = \frac{k \sigma_x}{\sigma_y}. \quad (2.19)$$

Отримане відношення служить для визначення вимірника тісноти зв'язку між змінними в разі їх лінійної залежності, і має назву коефіцієнта кореляції.

Якщо замість  $k$  підставити формулу для його обчислень  $k = \frac{\overline{xy} - \bar{x} \cdot \bar{y}}{\sigma_x \sigma_y}$ , то

матимемо



$$k = \frac{(\overline{xy} - \bar{x} \cdot \bar{y}) \sigma_x}{\sigma_x^2 \sigma_y} = \frac{\overline{xy} - \bar{x} \cdot \bar{y}}{\sigma_x \sigma_y}. \quad (2.20)$$

Коефіцієнт кореляції показує, на яку частину середнього квадратичного відхилення або  $\sigma_y$  змінюється функція  $y$ , якщо аргумент  $x$  збільшується (зменшується) на своє середньоквадратичне відхилення  $\sigma_x$ . Знак коефіцієнта кореляції співпадає із знаком коефіцієнта регресії, а його чисельне значення коливається в межах

$$-1 \leq r_{y/x} \leq 1. \quad (2.21)$$

Суттєвість коефіцієнта кореляції при заданому рівні значущості  $\alpha=0,05$  або 5% перевіряємо за умовою

$$\frac{|r_{y/x}| \cdot \sqrt{n}}{1 - r_{xy}^2} \geq t_a \geq 1.96, \quad (2.22)$$

де  $t_a$  визначається за умовою  $t_a = 1 - \alpha$ .

Кінцевим дослідженням парного рівняння є перевірка відповідності виведеного рівняння описуваному реальному процесу. Якщо отримано декілька рівнянь регресії, то кращим слід читати те з них, яке ближче до закономірності по суті досліджуваного процесу, що вивчається. Слід оцінити ступінь близькості результатів розрахунків по кожному з отриманих рівнянь до звітних даних. Ступінь оцінюється по залишковій теоретичній дисперсії  $\tilde{\sigma}_{y/x}^2$ , яка характеризує розкид досліджених точок кореляційного поля навколо теоретичної лінії регресії під впливом чинників, що не враховані в отриманому рівнянні.

Залишкова теоретична дисперсія визначається з рівняння

$$\tilde{\sigma}_{y/x}^2 = \frac{\sum_{i=1}^k \sum_{j=1}^{h_i} (y_{ij} - \tilde{y}_i)^2}{n - p}, \quad (2.23)$$

де  $n$  - обсяг вибірки;  $p$  - число параметрів рівняння;  $k$  - число інтервалів;  $h$  - число спостережень в  $i$ -му інтервалі;  $y_{ij}$  - значення досліджених даних

функціональної ознаки;  $\tilde{y}_i$  - розрахункове значення функціональної ознаки, обчислене за рівнянням регресії.

Якщо складено декілька рівнянь регресії і для кожного обчислена залишкова дисперсія, то з декількох рівнянь регресії, рівноцінних по суті, перевагу слід віддати тому, в якого залишкова дисперсія менше.

Для керування регресії з одним аргументом  $x$  залишкову дисперсію обчислюємо за формулою

$$\tilde{\sigma}_{y/x}^2 = \sigma_y^2 (1 - \eta_{y/x}^2). \quad (2.24)$$

Однією з ознак збіжності початкових даних з отриманим рівнянням служить нормальність розподілу відхилень досліджених даних від розрахункових

$(y_{ij} - \tilde{y}_i)$  зі середнім значенням, рівним нулю, і середнім квадратичним відхиленням, рівним  $\sigma_{y/x}$ . Чим ближче розподіл цих відхилень до вказаного нормального закону, тим краще узгоджуються вихідні дані з виведеним рівнянням регресії. Обчисливши всі відхилення  $(y_{ij} - \tilde{y}_i)$ , перевіряємо наявність нормального закону їх розподілу візуально і за критерієм згоди.

Таким чином, дослідження керування в тій або іншій формі полягає у визначенні його параметрів (із вирішення системи нормальних рівнянь), підстановці в отримане рівняння всіх вихідних значень аргументу, розрахунку відповідних значень  $\tilde{y}_{ij}$ , порівнянні розрахункових значень  $\tilde{y}_i$  із вихідними значеннями, обчисленні суми квадратів відхилень розрахункових значень від вихідних і визначення залишкової дисперсії  $\tilde{\sigma}_{y/x}^2$ . Найкраща форма визначається за найменшою залишковою дисперсією. За такою схемою визначаються форми зв'язку  $y$  по всіх аргументах.

Існує і критерій адекватності, запропонований Фішером [27], заснований на порівнянні залишкової теоретичної дисперсії  $\sigma_{y/x}^2$  і загальної дисперсії  $\sigma_y^2$ .

Розглядається відношення  $\frac{\sigma_{y/x}^2}{\sigma_y^2} = T_{розрах}$  та порівнюється з табличним при заданому рівні значущості і різних ступенях свободи.

Загальна дисперсія  $\sigma_y^2$  досліджених даних від їх середнього значення встановлюється з урахуванням числа ступенів свободи  $f = n - k$  :

$$\tilde{\sigma}_{y/x}^2 = \frac{\sum_{i=1}^k \sum_{j=1}^{h_i} (y_{ij} - \bar{y})^2 m_i}{n - k}, \quad (2.25)$$

де  $k$  – число інтервалів у вибіркових даних.

Залишкова теоретична дисперсія  $\tilde{\sigma}_{y/x}^2$  встановлюється як різниця розрахункових  $\tilde{y}$  і середніх інтервальних значень  $\tilde{y}_i$  з урахуванням числа ступенів свободи  $d_1 = k - p$  та  $d_1 = n - k$ , де  $p$  – число параметрів керування.

Якщо  $\bar{f}_{роз} \leq \bar{f}_{табл}$ , то при заданому рівні значущості складене рівняння регресії затверджується. Вірогідність помилки тим менше, чим більше рівень значущості.

У разі, коли чисельник  $\tilde{\sigma}_{y/x}^2$  менше знаменника  $\tilde{\sigma}_y^2$ , то міняємо їх місцями разом з відповідними ступенями свободи  $d_1 = k - p$  та  $d_1 = n - k$ , де  $p$ .

Правило складання системи нормальних рівнянь для відшукування  $\beta$ -коефіцієнтів формується таким чином: за коефіцієнти при невідомих  $\beta_1, \beta_2, \dots, \beta_p$  приймаються "внутрішні коефіцієнти" кореляції між чинниками-аргументами, а як вільні члени - "зовнішні коефіцієнти" кореляції між функцією і кожним з чинників.

Регресійну модель в більшості випадків розглядають як інструмент аналізу, планування і керування виробництвом. Звідси особливо строгі вимоги ставляться до надійності, адекватності й точності кожного коефіцієнта моделі. Якщо стандартна помилка знайденого коефіцієнта регресії перевершує його за абсолютною величиною, то не можна поручитися за достовірність не тільки того, на скільки одиниць свого найменування в середньому змінюється  $y$  при зміні  $X$

на одиницю свого вимірювання, але і за напрям впливу даного чинника-аргументу.

Точність визначення коефіцієнтів множинної регресії суттєво залежить від ступеня стійкості системи нормальних рівнянь або, інакше, від ступеня обумовленості кореляційної матриці.

Система нормальних рівнянь є добре обумовленою, якщо малим змінам коефіцієнтів відповідають малі (того ж порядку) зміни рішень. Інакше кажучи, має місце безперервний зв'язок між коефіцієнтами системи рівнянь і її коренями - корені системи стійкі при малих змінах її коефіцієнтів.

### **2.3. Оцінка ефективного використання та продажу електроенергії, отриманої від відновлювальних джерел енергії в системі MicroGrid**

Одним з напрямків розвитку інтелектуальної електроенергетичної системи з активно-адаптивної мережею постає концепція MicroGrid. Система MicroGrid найчастіше включає в себе джерела розподіленої генерації, накопичувачі енергії і локальних споживачів [40]. Важливою властивістю MicroGrid можна назвати те, що, незважаючи на функціонування в рамках розподільної системи, вони можуть автоматично переходити в ізолюваний режим роботи у разі аварій в мережі і відновлювати синхронну роботу з мережею після усунення аварії з підтриманням необхідної якості електричної енергії. Будучи автономними або підключеними до національної енергетичної мережі, об'єкти розподільної генерації розташовані в безпосередній близькості від споживачів (невеликих міст, сіл, заводів) і виробляти електроенергію «на місці», істотно знижуючи втрати при передачі по лініях і підвищуючи, таким чином, ККД. Електрична енергія, вироблена об'єктами розподільної генерації, буде безпосередньо залежати від попиту локальних споживачів, які в свою чергу будуть мати можливість коригувати поставки енергії відповідно до своїх потреб, що веде до підвищення їх ролі в керуванні енергосистемою.

Необхідність прогнозування електроспоживання в системах MicroGrid обумовлена економічними причинами. Основною причиною є поява ринка електроенергії, при цьому зростає важливість прогнозування енергоспоживання учасників даного ринку. Відхилення фактичного споживання від заявлених значень більше певного відсотка призводить до закупівлі електроенергії з балансуєчого ринку за більшою ціною. Відхилення в меншу сторону також карається сплатою недопоставленої електроенергії, яка визначається різницею між заявленим і фактичним споживанням за встановленими розцінками [54]. Складність прогнозу електроспоживання обумовлена необхідністю обліку багатьох факторів, що впливають на споживання та генерацію електричної енергії (температура навколишнього повітря; тривалість дня; день тижня; ступінь освітленості; переходи з зимового на літній час і назад; наявність екстраординарних подій; прогнози погодних умов; стан інших факторів, що впливають на зміну споживання відповідно до даних, отриманими в результаті обробки статистики споживання; плановане включенні або відключенні енергоємних виробництв) [54]. Вибір метода прогнозування є досить складним завданням, від рішення якої залежить як підвищення надійності системи, так і уникання невідповідності попиту і пропозиції електроенергії. Проблема привабливості інтелектуальної енергетики для споживачів полягає в принциповому розриві між потребою інтелектуальної енергетики в активній участі широкого класу споживачів в управлінні нею для забезпечення гнучкості та підвищення ефективності енергосистеми і нинішньою поведінкою споживачів, які не зацікавлені в участі в управлінні енергетикою і не розуміють, навіщо і як вони могли б це робити.

Очевидно, що точність прогнозу визначається кількістю параметрів, що беруться до розгляду, а також глибиною ретроспективи часового ряду цих параметрів. Для прикладу було проведено дослідження часового ряду, що містить дані про ціну на електроенергію мережі, зелений тариф [37] та обчислений прибуток від продажу надлишків електроенергії, згенерованої

альтернативними джерелами MicroGrid. У табл. 2.1 наведено приклад такого часового ряду за 2010-2018 рр. Ретроспективні статистичні дані надалі екстраполуються на майбутні періоди.

Таблиця 2.1

	Ціна за електроенергію мережі, грн/кВт*год ( $x_1$ )	Зелений тариф, грн/кВт*год ( $x_2$ )	Прибуток за продану електроенергію, грн/кВт*год ( $y$ )
2010	0,60	14,11	689,5
2011	0,72	13,12	701,4
2012	0,84	12,19	684,4
2013	0,95	11,19	650,3
2014	1,07	10,70	648,8
2015	1,31	10,06	645,6
2016	1,42	10,04	643,0
2017	1,50	10,01	642,2
2018	1,52	10,00	642,1

Спираючись на отримані дані в табл. 2.2, можна зробити прогноз, що ціна електроенергії в мережі на 2019 р. становитиме 1,55 грн за 1 кВт/г.

Таблиця 2.2

$y$	$x_1$	$x_2$	$(y - \bar{y})^2$	$(x_1 - \bar{x}_1)^2$	$(x_2 - \bar{x}_2)^2$
689,50	0,60	14,11	823,052	0,253	8,07
701,40	0,72	13,12	1647,458	0,146	3,43
684,40	0,84	12,19	556,436	0,069	0,85
650,30	0,95	11,19	110,484	0,023	0,01
648,80	1,07	10,70	144,267	0,001	0,32
645,60	1,31	10,06	231,378	0,042	1,46
643,00	1,42	10,04	317,236	0,100	1,51
642,20	1,50	10,01	346,374	0,157	1,58
642,10	1,52	10,00	350,106	0,174	1,61

В табл. 2.2  $\bar{y}$  - середнє арифметичне значення прибутку за продану електроенергію (660.81 грн/кВт\*год),  $\bar{x}_1$  - середнє арифметичне значення фактора  $x_1$  (1,10 грн/кВт\*год),  $\bar{x}_2$  - середнє арифметичне значення фактора  $x_2$  (11,27 грн/кВт\*год).

Часові ряди параметрів підлягають кореляційно-регресійному аналізу. При цьому необхідно встановити наявність кореляційних взаємозв'язків між параметрами, які розглядаються як випадкові величини та відповідно підлягають статистичній оцінці. Кореляційні моменти (коваріації) характеризують наявність або відсутність зв'язку між випадковими величинами  $x_1$ ,  $x_2$  та  $y$  з табл.1 за формулами (2.26) та (2.27):

$$K_{yx} = \frac{\sum_{i=1}^n (y_i - \bar{y})(x_i - \bar{x})}{n}, \quad (2.26)$$

де  $K_{yx_1} = -6,4624$ ;  $K_{yx_2} = 30,3868$ ,

$$K_{x_1x_2} = \frac{\sum_{i=1}^n (x_i - \bar{x}_1)(x_i - \bar{x}_2)}{n} = \frac{-4,0185}{9} = -0,4465, \quad (2.27)$$

де  $\bar{y}$  - середнє арифметичне значення прибутку за продану електроенергію (660.81 грн/кВт\*год),  $\bar{x}_1$ - середнє арифметичне значення фактора  $x_1$  (1,10 грн/кВт\*год),  $\bar{x}_2$  - середнє арифметичне значення фактора  $x_2$  (11,27 грн/кВт\*год),  $n$  – кількість значень (9).

Як бачимо, кореляційний момент у всіх випадках розглянутого прикладу відмінний від нуля – отже, кореляційний зв'язок існує між усіма змінними. Нормоване значення коефіцієнта коваріації називається коефіцієнтом кореляції.

Для перевірки наявності кореляції парного лінійного зв'язку використовують лінійний коефіцієнт парної кореляції, який є мірою пропорційності ступеня залежності явищ, показником інтенсивності лінійного зв'язку. Для розглянутого випадку коефіцієнти кореляції приймають значення:

$$r_{yx_1} = \frac{\overline{x_1 \cdot y} - \bar{x}_1 \cdot \bar{y}}{(\overline{x_1^2} - \bar{x}_1^2) \cdot (\overline{y^2} - \bar{y}^2)} = 0,88; \quad (2.28)$$

$$r_{yx_2} = \frac{\overline{x_2 \cdot y} - \bar{x}_2 \cdot \bar{y}}{(\overline{x_2^2} - \bar{x}_2^2) \cdot (\overline{y^2} - \bar{y}^2)} = 0,94; \quad (2.29)$$

$$r_{x_1x_2} = \frac{\overline{x_1 \cdot x_2} - \bar{x}_1 \cdot \bar{x}_2}{(\overline{x_1 - \bar{x}_1})^2 \cdot (\overline{x_2 - \bar{x}_2})^2} = 0,94. \quad (2.30)$$

Коефіцієнт кореляції приймає чисельні значення в проміжку від -1 до +1. Якщо значення позитивне, це свідчить про прямий зв'язок, негативне - про зворотній.

Коефіцієнт кореляції перевіряється шляхом знаходження t-критерію Стюдента [5]. Визначимо значення критерія t. Потім перевіримо на відсутність зв'язку між прибутком за продажу електроенергії та факторами.

$$t_{\text{роз}(yx)} = \frac{|r|\sqrt{n-2}}{\sqrt{1-r^2}} = 4,866, \quad (2.31)$$

$t_{\text{роз}(yx_1)} = 4,866$ ;  $t_{\text{роз}(yx_2)} = 7,059$ ;  $t_{\text{роз}(x_1x_2)} = 7,339$ ;  $t_{\text{табл}} = 1,8331$  з рівнем імовірності 0,90.

Обчислене за формулою (6) значення  $t_{\text{роз}}$  порівнюється з  $t_{\text{табл}}$ , який отримують за таблицею Стюдента [51]. У разі, якщо  $|t_{\text{роз}}| > |t_{\text{табл}}|$ , зв'язок вважають суттєвим. Оскільки  $|4,866| > |1,8331|$ ,  $|7,059| > |1,8331|$  та  $|7,339| > |1,8331|$ , це означає, що зв'язок є суттєвим для будь-якої з пар розглянутих коефіцієнтів табл.2.1.

Регресійний аналіз є кількісним методом визначення виду математичної функції в причинно-наслідкової залежності між прибутком за продажу електроенергії і факторами, які впливають на нього. Тому необхідно зробити прогноз результативного показника (y), побудовою рівняння регресії. Лінійне рівняння множинної регресії має вигляд:

$$\overline{y_{x_i}} = a_0 + a_1x_1 + a_2x_2 + \dots + a_ix_i + \dots + a_nx_n, \quad (2.32)$$

де  $x_i$  - факторні ознаки (в розглянутому випадку їх дві –  $x_1$  та  $x_2$ );  $a_i$  - невідомі параметри рівняння (коефіцієнти регресії).

Коефіцієнти регресії  $a_i$  показують, наскільки змінюється значення прибутку за продажу електроенергії у при збільшенні її факторної значення  $x_i$



на одиницю при фіксованому значенні інших факторних ознак. Вільний член рівняння  $a_0$  показує усереднений вплив на результативний показник всіх неврахованих факторів.

Для розглянутого випадку рівняння множинної регресії з двома факторами має вигляд:

$$\overline{y_{x_i}} = a_0 + a_1x_1 + a_2x_2. \quad (2.33)$$

Для знаходження коефіцієнтів лінійної двофакторної моделі необхідно вирішити систему рівнянь з трьома невідомими параметрами  $a_0$ ,  $a_1$ ,  $a_2$ :

$$\begin{cases} a_0n + a_1 \sum x_1 + a_2 \sum x_2 = \sum y \\ a_0 \sum x_1 + a_1 \sum x_1^2 + a_2 \sum x_1x_2 = \sum x_1y \\ a_0 \sum x_2 + a_1 \sum x_1x_2 + a_2 \sum x_2^2 = \sum x_2y \end{cases}, \quad (2.34)$$

де  $a_1$  та  $a_2$  є коефіцієнтами регресії,  $a_0$  - вільним членом рівняння,  $n$  – кількість значень (9).

Отримаємо значення коефіцієнтів:

$$a_0 = 492,34; \quad a_1 = 1,40; \quad a_2 = 14,81.$$

Остаточне рівняння лінійної регресії має вигляд:

$$\overline{y_{x_i}} = 492,34 + 1,40x_1 + 14,81x_2.$$

Розглянемо економічне трактування результатів побудованої двофакторної регресійної моделі. Лінійна регресія виражає залежність результативного показника – прибутку проданої електроенергії ( $y$ ) – від ціни на зелений тариф ( $x_1$ ) та ціни на електроенергію в мережі ( $x_2$ ). Коефіцієнти рівняння підтверджують наявність впливу всіх факторів на результативний показник. У нашому випадку, за один рік ціна на електроенергію в мережі збільшується в середньому на 1,40 грн/кВт\*год при збільшенні ціни зеленого тарифу на 1 грн/кВт\*год при незмінності ціни на електроенергію в мережі. Прибуток від проданої електроенергії за зеленим тарифом збільшується в середньому на 14,81 грн/кВт\*год при збільшенні ціни на електроенергію в мережі на 1 грн/кВт\*год при незмінності ціни на зелений тариф.

Отримані дані дозволяють прогнозувати економічні параметри локального ринку MicroGrid та відповідно планувати обсяги закупівель та продажів електроенергії його учасниками.

Таким чином, проведений кореляційно-регресійний аналіз підтверджує існування статистичного взаємозв'язку між прибутком від реалізації електроенергії за зеленим тарифом, цінами на зелений тариф та цінами на електроенергію в мережі. Обчислення коефіцієнтів двофакторної регресійної моделі встановлює рівень залежності між параметрами та дозволяє прогнозувати економічні параметри локального ринку MicroGrid для планування обсяги закупівель та продажів електроенергії його учасниками.

#### **2.4. Лінійна регресія в програмному забезпеченні на мові C#**

Мета рішення задачі лінійної регресії у програмному забезпеченні - передбачити значення числової змінної на основі значень однієї або більше числових змінних-предикторів.

Прогнозована змінна називається залежною (dependent variable), а змінні-предиктори - незалежними (independent variables). При наявності тільки однієї змінної-предиктора цей метод називають простою лінійною регресією. Коли є дві або більше змінних-предикторів цей метод називають множинною лінійною регресією (multiple, або multivariate, linear regression). У данному випадку використовується множинна лінійна регресія. На рис. 2.5 представлено скріншот програми написаний на мові програмування C#.

Програма показує кількість накопичених залишків електроенергії на основі отриманих даних від пристрою збору даних. Демонстрація починається з генерації елементів даних. У першій колонці вводяться дані енерговитрат за день (кВт/год). Друга колонка показує дані дня тижня від 1 до 7, третя приймає значення сезону (1 – літо; -1 – зима; 0 – весна та осінь), а четверта приймає значення даних накопичених залишків електроенергії за день.

```

C:\Users\Orlov\Диплом Магістра\Regression analysis\Regression analysis.exe
*** Програма оцінки кількості накопиченої енергії за допомогою регресійного аналізу ***
*** Отримання даних за тиждень ****
-Готово-
Кіл-сть витрач. елек-ргії за день - День тижня - Сезон - Залишок:
12,00  1,00  1,00  19,28
18,00  2,00  1,00  24,82
13,00  3,00  1,00  19,13
17,00  4,00  1,00  24,98
13,00  5,00  1,00  27,70
17,00  6,00  1,00  31,41
17,00  7,00  1,00  36,14
*** Створення матриці проектування з даних ***
-Готово-
Матриці проектування:
1,00  12,00  1,00  1,00  19,28
1,00  18,00  2,00  1,00  24,82
1,00  13,00  3,00  1,00  19,13
1,00  17,00  4,00  1,00  24,98
1,00  13,00  5,00  1,00  27,70
1,00  17,00  6,00  1,00  31,41
1,00  17,00  7,00  1,00  36,14
*** Пошук коефіцієнтів з використанням інверсії ***
-Готово-
Коефіцієнти:
6,4620  0,7010  2,2578  0,0000
Обчислення R-квадрату
R-квадрат = 0,8817
Прогнозування залишку для
Кіл-сть витрач. елек-ргії за день = 12
День тижня = 1 (понеділок)
Сезон = 1 (літо)
Прогнозований залишок = 19,24
*** Кінець програми ***

```

Рис 2.5 Скріншот програми для розв'язку лінійної регресії

Після генерації синтетичних даних програма створює на їх основі так звану матрицю плану (design matrix). Це матриця з даними, в якій всі значення в колонці приймає значення одиниці. Існують різні алгоритми, які можна застосовувати для лінійної регресії; деякі з них можуть використовувати

матрицю з вихідними даними (raw data matrix), тоді як інші працюють з матрицею плану.

Після створення такої матриці програма знаходить значення для чотирьох коефіцієнтів (6.4620, 0.7010, 2.2578, 0). Ці коефіцієнти називають b-значеннями, або бета-значеннями. Перше значення, 6.4620, в статистиці називається вільним членом (intercept). Це константа, не пов'язана ні з якими змінними-предикторами. Значення інших коефіцієнтів (0.7010, 2.2578, 0) відносяться до енерговитрат, доби та сезону відповідно.

Програма для прогнозування за допомогою лінійної регресії здійснює множення значень предикторів на відповідні коефіцієнти з подальшим додаванням результатів. Зауважимо, що значення вільного члена (в цьому прикладі - 6.4620) можна вважати коефіцієнтом, пов'язаним із змінною-предиктором, значення якої завжди дорівнює 1.

Суть задачі лінійної регресії полягає в обчисленні значень коефіцієнтів, використовуючи вихідні дані або, що еквівалентно, матрицю плану. У програмі застосовується метод, званий зверненням матриці в замкнутій формі (closed form matrix inversion), також відомий як звичайний метод найменших квадратів (ordinary least squares method) [47]. Альтернативні методи знаходження значень коефіцієнтів включають ітераційний зважений метод найменших квадратів (iteratively reweighted least squares), оцінку за методом максимальної правдоподібності (maximum likelihood estimation), гребеневу регресію (ridge regression), градієнтний спуск (gradient descent) і кілька інших.

Програма, перш ніж приступити до прогнозування, обчислює показник, званий значенням R-квадрата (R-squared value), або коефіцієнтом детермінації (coefficient of determination). R-квадрат - це значення між 0 і 1, яке описує, наскільки добре модель прогнозування підходить для вихідних даних. Іноді це висловлюють у вигляді «процентної частки відхилення, що пояснюється моделлю». У вільному тлумаченні можна сказати так: чим ближче R-квадрат до 1, тим краще модель прогнозування. У демонстраційній програмі це значення

становить 0.8817, або 82%, і для реальних даних воно вважалося б порівняно високим (гарним).

#### 2.4.1. Рішення рівняння найменших квадратів

Якщо в задачі лінійної регресії є  $n$  змінних-предикторів, тоді потрібно знайти  $n + 1$  значень коефіцієнтів - по одному на кожен предиктор плюс вільний член (intercept). У програмі використовується базовий метод для знаходження значень коефіцієнтів.

$$\beta = (X^T \cdot X)^{-1} \cdot X^T \cdot Y,$$

$\beta$  - представляє значення коефіцієнтів. В математиці коефіцієнти  $X$ ;  $X^T$  та  $Y$  представляють об'єкти з кількома значеннями (матриці або масиви / вектори), а не прості скалярні величини. Коефіцієнт  $X$  представляє матрицю плану. Коефіцієнт  $X^T$  означає транспонування матриці плану. Коефіцієнт  $Y$  - вектор-колонка (матриця з одним стовпцем) значень залежної змінної. Тому знаходження значень коефіцієнтів насправді вимагає розуміння матричних операцій.

Схеми на рис. 2.6 ілюструють транспонування, множення і зобернення матриць. При транспонуванні матриця перетвориться в матрицю  $3 \times 2$ , де рядки вихідної матриці стали колонками у транспонованій матриці.

Якщо помножити матрицю розміром  $(n \times m)$  на матрицю розміром  $(m \times p)$ , результатом буде матриця розміром  $(n \times p)$ . Наприклад, множення матриць  $3 \times 4 * 4 \times 2$  дає матрицю  $3 \times 2$ .

Існують різні алгоритми, які можна застосувати для обернення матриці. У програмі використовується метод розкладання Дулітла (Doolittle's decomposition)[47].

Таким чином. Завдання лінійної регресії з  $n$  змінними-предикторами включає знаходження значень для  $n + 1$  коефіцієнтів. Це можна зробити, використовуючи матричні операції: транспонування, множення і звернення, які є достатньо простими в розумінні і реалізації.

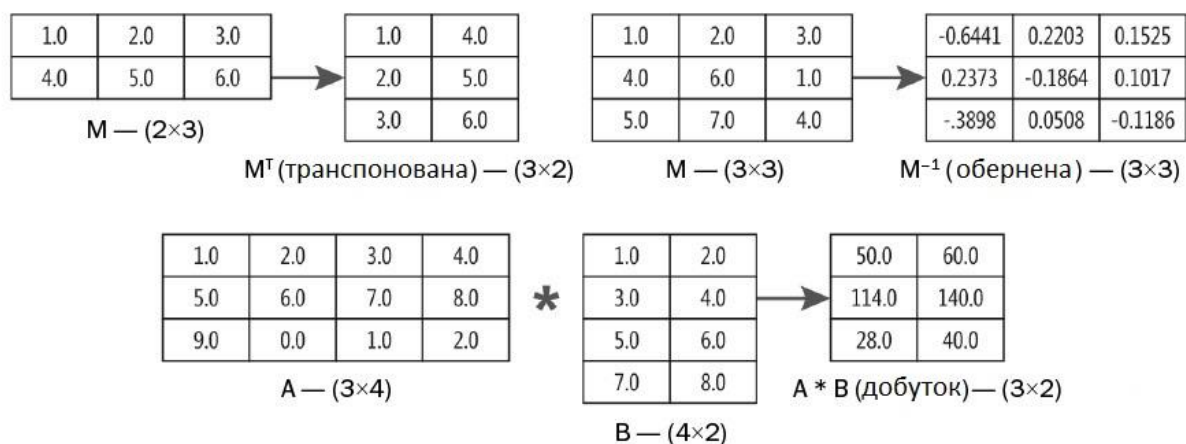


Рис 2.6 Три матричні операції, використовувані при знаходженні коефіцієнтів лінійної регресії

### 2.4.2. Структура демонстраційної програми

Щоб створити програму, у середовищі розробки Visual Studio було вибрано шаблон C # Console Application.

Вся керуюча логіка програми знаходиться в Main. Демонстраційна програма використовує підхід на основі статичних методів, а не об'єктно-орієнтоване програмування.

```
using System;
namespace LinearRegression
{
    class LinearRegressionProgram
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Begin linear regression demo");
            // Генерація синтаксичних даних.
            // Створення матриці плану.
            // Знаходимо коефіцієнти лінійної регресії.
            // Підраховуємо R-квадратне значення.
            // Робимо оцінку.
            Console.WriteLine("End linear regression demo");
            Console.ReadLine();
        }
        static double Income(double x1, double x2,
```

```

    double x3, double[] coef) { .. }
static double RSquared(double[][] data,
    double[] coef) { .. }
static double[][] DummyData(int rows,
    int seed) { .. }
static double[][] Design(double[][] data) { .. }
static double[] Solve(double[][] design) { .. }
static void ShowMatrix(double[][] m, int dec) { .. }
static void ShowVector(double[] v, int dec) { .. }
// -----
static double[][] MatrixTranspose(double[][] matrix)
    { .. }
static double[][] MatrixProduct(double[][] matrixA,
    double[][] matrixB) { .. }
static double[][] MatrixInverse(double[][] matrix)
    { .. }
}
} // ns

```

Повний код програми знаходиться у додатку 1.

Метод Income повертає прогнозований залишок енергії на основі вхідних параметрів, використовуючи масив значень коефіцієнтів. Метод RSquared повертає R-квадратичне значення моделі на основі даних і коефіцієнтів. Метод DummyData генерує синтетичні дані, що застосовуються в демонстрації.

Метод Design приймає матрицю даних і повертає доповнену матрицю плану, колонка яка містить значення одиниці. Метод Solve приймає матрицю плану і використовує матричні операції для знаходження коефіцієнтів лінійної регресії.

Більшу частину основної роботи бере на себе набір статичних методів, які виконують матричні операції. У демонстрації визначена гранично проста матриця - масив масивів. Альтернатива полягає в створенні класу Matrix, але, на мій погляд, цей підхід тут невиправдано складний.

Метод MatrixTranspose повертає транспоновану матрицю, метод MatrixProduct - результат перемноження двох матриць, а метод MatrixInverse - обернену матрицю. У демонстрації багато допоміжних методів. Зокрема, метод

MatrixInverse викликає допоміжні методи MatrixDuplicate, MatrixDecompose і HelperSolve.

Центральне місце в програмі, яка демонструє лінійну регресію, займає метод Solve. Визначення цього методу починається з частини коду наступного вигляду:

```
static double[] Solve(double[][] design)
{
    int rows = design.Length;
    int cols = data[0].Length;
    double[][] X = MatrixCreate(rows, cols - 1);
    double[][] Y = MatrixCreate(rows, 1);
    ...
}
```

Єдиним вхідним параметром є матриця плану (design). Можна подумати про альтернативний підхід: передавати матрицю з вихідними даними, а потім змушувати Solve викликати допоміжний метод Design, щоб отримати матрицю плану. Допоміжний метод MatrixCreate виділяє простір під матрицю з вказаною кількістю рядків і стовпців і повертає її. Локальна матриця X містить значення незалежних змінних (з провідним значенням 1.0). Локальна матриця Y має тільки одну колонку і зберігає значення залежної змінної (в нашому прикладі – залишок електроенергії).

Потім матриці X і Y заповнюються, використовуючи значення в матриці плану:

```
int j;
for (int i = 0; i < rows; ++i)
{ for (j = 0; j < cols - 1; ++j)
  { X[i][j] = design[i][j]; }
  Y[i][0] = design[i][j]; }
```



Зауважимо, що змінна-індексу  $j$  оголошується поза вкладених циклів `for`, щоб її можна було використовувати для заповнення матриці  $Y$ . Маючи матриці  $X$  і  $Y$ , можна знайти коефіцієнти лінійної регресії відповідно до алгоритму:

```
...
double[][] Xt = MatrixTranspose(X);
double[][] XtX = MatrixProduct(Xt, X);
double[][] inv = MatrixInverse(XtX);
double[][] invXt = MatrixProduct(inv, Xt);
double[][] mResult = MatrixProduct(invXt, Y);
double[] result = MatrixToVector(mResult);
return result; } // Solve
```

У демонстрації матриця  $X$  має розмір  $7 \times 4$ , тому її транспонована версія,  $X_t$ , отримує розмір  $4 \times 7$ . Результат перетворення  $X_t$  і  $X$  має розмір  $4 \times 4$ , так само, як і обернена матриця,  $inv$ . В цілому, для завдання лінійної регресії з  $n$  незалежними змінними-предикторами при використанні методу звернення матриці потрібно знайти звернення матриці розміром  $(n + 1) \times (n + 1)$ . Тобто метод звернення не годиться для задач лінійної регресії з величезним числом змінних-предикторів.

Результат множення оберненої матриці  $4 \times 4$  і транспонованої матриці  $4 \times 7$  (в коді це змінна  $invXt$ ) має розмір  $4 \times 7$ . Результат множення  $invXt$  і матриці  $Y$  розміром  $7 \times 1$  (в коді це  $mResult$ ) виходить розміром  $4 \times 1$ . Ці значення є потрібними коефіцієнтами. Для зручності значення в матриці  $Y$  з однією колонкою переміщуються в звичайний масив викликом допоміжного методу `MatrixToVector`.

Як вже зазначалося, показник  $R$ -squared ( $R$ -квадрат, або коефіцієнт детермінації) є мірою того, наскільки добре точки реальних даних відповідають обчисленої лінії регресії. У математичних термінах  $R$ -квадрат визначається як  $R^2 = 1 - (SS_{res} / SS_{tot})$ . Член  $SS_{res}$  зазвичай називається залишковою сумою квадратів (residual sum of squares). Це сума квадратів різниці (squared differences)

між реальними і передбаченими  $Y$ -значеннями. Член  $SStot$  - це загальна сума квадратів (total sum of squares). Цей член є сумою квадратів різниці між кожним реальним  $Y$ -значенням і середнім всіх реальних  $Y$ -значень.

Показник  $R$ -squared в лінійної регресії також називається коефіцієнтом детермінації і має відношення (але відрізняється від нього) до іншого статистичного показника з назвою « $r$ -squared» («малий  $r$ -квадрат»). Інтерпретація  $R$ -квадрата не зовсім однозначна і залежить від предметної області конкретного завдання. У природних і суспільних науках, де дані, як правило, заплутані і неповні, значення  $R$ -квадрата від 0.6 і вище частіше трактується як досить гарне.

Існує альтернативна міра відхилень, під назвою «скоригований  $R$ -квадрат» (adjusted  $R$ -squared). Цей показник враховує кількість змінних-предикторів і число елементів даних. Для більшості цілей застосування звичайного значення  $R$ -квадрата цілком достатньо, щоб отримати уявлення про прогностичну якість моделі лінійної регресії.

### **Висновки до другого розділу**

Прогнозування залишку електроенергії можна досягти багатьма способами, використовуючи їх переваги та компенсуючи недоліки. Звичайно, вибір оптимального алгоритму залежить від мети, а також від аналітики даних. З одного боку, немає універсальних та оптимальних методів для всіх програм, а прогнозування попиту на енергію є лише одним із прикладів такої ситуації. З іншого боку, на практиці використовувати складні методи для виконання завдань не доцільно.

Використання лінійного регресійного аналізу (для всіх розглянутих прогнозних періодів) дозволило отримати більшу точність прогнозу ніж статистичний метод, який використовувався як еталонний. Метод лінійної регресії показав малу похибку результатів які були отримані завдяки великій кількості підрахунків за різний період часу та різними значеннями факторів.

Суттєвою перевагою даного метода є наявність інтерпретації, а також обчислювальна простота.

Ефективність даного підходу прогнозування зазвичай залежить від багатьох факторів, в тому числі правильності і точності моделей попиту на електроенергію, наявність нестоціонарних змін, пов'язаних зі святами або погодними умовами протягом певного періоду часу і т.д.

Метод квадратичної регресії, показав схожі результати по точності прогнозування, але даний варто розглядати у випадках, коли застосовуються декілька різноманітних методик прогнозування. Всі прогнози можуть бути вдосконалені, за допомогою більш складних законів, що описують фактори, зміною кількості історичних даних (періодів), які використовуються для прогнозування, використання інших методів або більш ретельного вибіру параметрів.

### РОЗДІЛ 3. БАЗИ ДАНИХ ТА СИСТЕМИ КЕРУВАННЯ БАЗАМИ ДАНИХ

База даних (БД) – це упорядкований набір логічно взаємопов'язаних даних, що використовується спільно, та призначений для задоволення інформаційних потреб користувачів.

Існує два основних типи баз даних: SQL і NoSQL – реляційна і нереляційна база даних відповідно [31]. Різниця полягає в тому, як вони побудовані, який тип інформації зберігають і як саме.

Система керування базами даних (СКБД) - це комплекс програмних і мовних засобів, необхідних для створення баз даних, підтримання їх в актуальному стані та організації пошуку в них необхідної інформації.

СКБД відповідають за:

- пошук потрібних даних;
- фізичне розміщення даних і їх описів;
- відновлення і оновлення баз даних відповідно до змін у реальному світі (підтримка актуального стану);
- захист даних від злому, некоректних змін і забороненого доступу;
- регулювання одночасних запитів до бази від декількох користувачів.

Розглянемо типи баз даних більш детально у розділі задачі програмування залишків електричної енергії від ВДЕ у MicroGrid.

#### 3.1. Реляційні бази даних

Реляційна база даних - це база даних, побудована на основі реляційної моделі, тобто БД, що має табличний спосіб зберігання даних, що задається набором однорідних таблиць.

Реляційна база даних створюється й потім управляється за допомогою спеціальних засобів — реляційних систем керування базами даних (РСУБД).

Історично РБД діляться на:

- РБД (РСБД), створені для дуже великих (більше 1 Гбайт) баз даних архітектури «клієнт-сервер». Перші розробки виконані для більших комп'ютерів IBM, у яких використовується мова SQL;
- РБД (РСБД), створені спеціально для ПК типу dbase, у яких архітектура така, що база й користувач перебувають на одному комп'ютері.

У цей час намітилася тенденція їх зближення. Так, у СБД другого типу вводиться мова SQL, що дозволяє взаємодія БД різного типу.

Основні типи SQL-даних:

- INTEGER - тут зберігаються цілі числа, як позитивні, так і негативні. Наприклад: 2, 45, -16 і 23989.
- FLOAT - тут зберігаються числа, в яких необхідно вказувати десяткові знаки. Наприклад: 2.5, -.664, 43.8882 або 10.00001.
- DATETIME - зберігає дату і час у форматі PPPP-MM-ДД ГГ:XX:СС
- VARCHAR - тут зберігається обмежена кількість текстових або одиночних символів.
- BLOB - тут зберігаються двійкові дані, відмінні від тексту, наприклад, завантаження файлів.

Популярні бази даних SQL і реляційні системи керування базами даних (РСКБД) [29]:

- MS SQL - найпопулярніша база даних з відкритим вихідним кодом, відмінна для сайтів CMS та блогів.
- Oracle - об'єктно-реляційна СКБД, написана на мові C ++. Це повний набір сервісів, що забезпечує високий рівень обслуговування та надійність. Oracle також випустила базу даних Oracle NoSQL.
- IBM DB2 - це сімейство продуктів IBM з серверних баз даних, створені для обробки передових аналітичних "великих даних".

- Sybase - реляційна модель бази даних для серверів, яка в основному використовується в ОС Unix, яка була першою на рівні підприємства на базі СУБД для Linux.
- MS SQL Server - Microsoft-розроблена РСКБД для баз даних на рівні підприємства, що підтримує як SQL, так і NoSQL-архітектури.
- Microsoft Azure - платформа хмарних обчислень, яка підтримує будь-яку операційну систему, і дозволяє зберігати, обчислювати та масштабувати дані в одному місці.
- MariaDB – покращена версія MS SQL.
- PostgreSQL - об'єктно-реляційна СКБД на рівні підприємства, що використовує процедурні мови, такі як Perl і Python, в додаток до коду SQL-рівня.

Принцип роботи реляційної бази даних наведено на рис. 3.1.

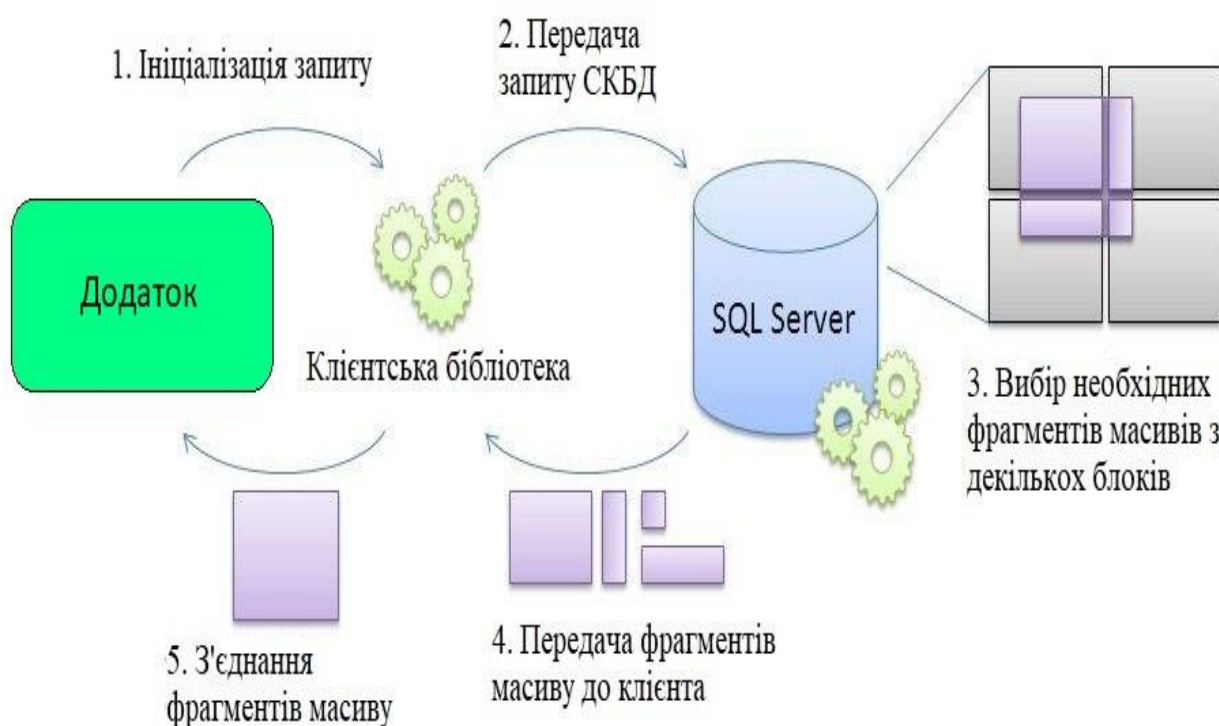


Рис 3.1 Принцип роботи реляційної бази даних

### 3.2. Теорема ACID

В області комп'ютерних наук ACID (Atomicity, Consistency, Isolation, Durability) являє собою набір особистих властивостей транзакцій бази даних, призначених для гарантії дійсності у разі помилок, збоїв живлення і т. п. Транзакція - це послідовність операцій з базою даних, яка задовольняє властивостям ACID і, таким чином, може сприйматися як єдина логічна операція над даними. ACID і BASE представляють дві філософії спектра узгодженості-доступності. Властивості ACID зосереджені на узгодженості та виборі спектра. Сучасні великомасштабні глобальні системи, включаючи хмарні, використовують обидва варіанти.

У 1983 році Андреас Рейтер і Тео Хердер придумали аббревіатуру ACID як скорочення для Atomicity, Consistency, Isolation і Durability, ґрунтуючись на більш ранній роботі Джима Грея, який перераховував Atomicity, Consistency і Durability, але залишив без ізоляції, коли характеризує концепцію транзакції. Ці чотири властивості описують основні гарантії парадигми транзакцій, які вплинули на багато аспектів розробки в системах баз даних:

- Валентність (Atomicity): дана властивість гарантує, що будь-яка транзакція не буде зафіксована в системі частково. Гарантовано будуть виконані всі її підоперації, або не виконано жодної з них. Оскільки на практиці неможливо одночасно і валентно (атомарно) виконати всю послідовність операцій у середині транзакції, вводиться поняття «відкату»: якщо не вдається повністю завершити транзакцію, результати всіх її дій будуть скасовані і система повернеться у початковий стан.
- Узгодженість (Consistency): транзакція, що досягає свого завершення фіксує свої результати та зберігає узгодженість бази даних. Іншими словами, кожна вдала транзакція по призначенню фіксує тільки допустимі результати. Ця умова є необхідною для підтримки четвертої властивості.

В банківській системі може існувати вимога рівності суми, що списується з одного рахунку, та зараховується на інший. Це бізнес-правило і воно не може бути гарантоване тільки перевітками цілісності, його повинні дотримуватись програмісти при написанні коду транзакцій. Якщо будь-яка транзакція обробляє списання, але не зробить зарахування, то система приймає некоректний стан і властивість узгодженості буде порушено.

Зауважемо, що в ході виконання транзакції узгодженість не потрібна. Списання і зарахування відбуваються, швидше за все, двома окремими підопераціями і між їх виконанням в транзакцією буде видно неузгоджений стан системи. Однак, при виконанні вимоги ізоляції ніяким іншим транзакціям ця неузгодженість не буде відображена. При цьому, атомарність гарантує, що транзакція або буде повністю завершена, або жодна з операцій транзакції не буде виконана. Тим самим ця проміжна неузгодженість є прихованою.

- Ізольованість (Isolation): під час виконання транзакції паралельні транзакції не повинні впливати на її результат. Ізольованість - обов'язкова вимога, тому в реальних БД існують режими, які не повністю ізолюють транзакцію.

- Стійкість (Durability): незалежно від проблем на нижніх рівнях зміни, зроблені транзакцією, яка успішно завершена, повинні залишитися збереженими після повернення системи в роботу. Іншими словами, якщо користувач отримав підтвердження від системи, що транзакція виконана, він може бути впевнений, що зроблені ним зміни не будуть скасовані через збій в системі.

Як приклад збій стійкості може відобразитися в такій ситуації. Транзакція передає значення 10и з точки А в точку В. Спочатку вона видаляє 10 з точки А, потім додає 10 до точки В. На цьому етапі користувачу повідомляється, що транзакція пройшла успішно, однак зміни як і раніше знаходяться в черзі в очікуванні дискового буфера бути прив'язаним до диску. Збій живлення і зміни губляться. Користувач передбачає (зрозуміло), що зміни зберігаються.



### 3.3. Нереляційні бази даних

Нереляційна база даних - база даних, яка забезпечує механізм зберігання та отримання даних, відмінний від підходу таблиць-відношень в реляційних базах даних. Подібні бази даних існували вже в другій половині 1960-х років, але тоді вони ще не здобули гучне ім'я «NoSQL», одержане після сплеску популярності на початку 21-го століття, що був спричинений потребами Web 2.0 компаній, такими як Facebook, Google, та Amazon.com. Нереляційні бази даних все більше і більше використовуються в задачах із застосуванням великих даних та real-time[en] web-застосунках. NoSQL системи також називають «Not only SQL» для підкреслення того, що вони можуть підтримувати SQL-подібну структуру та мову запитів.

Мотиви цього підходу включають: простоту дизайну схеми БД, значно спрощене горизонтальне масштабування на кластери (що є проблемою для реляційних баз даних), і тонкий контроль над доступністю. Структури даних, що використовуються в NoSQL (такі як ключ-значення, сховище з широким стовпчиком, граф, документ) є відмінними від тих, що використовуються за замовчуванням в реляційних базах, що робить тим самим деякі операції над даними значно швидшими на NoSQL. Точна відповідність використання NoSQL бази даних залежить від проблем, які треба вирішити. Іноді структури даних, які використовуються в NoSQL базах можуть розглядатись як більш гнучкі ніж таблиці реляційних моделей.

Структури даних, що використовуються в NoSQL, відрізняються від тих, що використовуються в реляційних базах даних. Точна відповідність використання реляційних баз даних залежить від проблеми, яку необхідно вирішити [30].

Кожне рішення в рамках технології NoSQL використовує свою власну модель. Ці моделі поділяються на чотири категорії: ключ-значення, документ,

сімейство колонок і граф. Перші три моделі мають загальну властивість, яку називають агрегатною орієнтацією (aggregate orientation).

Реляційна модель отримує інформацію, яку необхідно зберегти, і розділяє її на частини (рядки). Вони зберігають набір значень, тому не може містити запис, список значень або іншу частину. Ця простота утворює основу реляційної моделі і дозволяє інтерпретувати всі операції як операції над частинами та їх повернення.

Агрегатна орієнтація дотримується іншого підходу. Вона враховує необхідність оперувати даними, що мають більш складну структуру, ніж набір частин. Її можна описати в термінах складних записів, яка може містити списки і інші структури записів. Баз даних типу "ключ-значення", документ і сімейство стовпців можуть містити складні записи – агрегати.

У предметно-орієнтованому проектуванні агрегатом називають колекцію пов'язаних об'єктів, яка інтерпретується як єдине ціле. Зокрема, вона являє собою одиницю для маніпулювання даними та керування їх узгодженістю. Зазвичай агрегати модифікуються за допомогою атомарних операцій і взаємодіють зі сховищем даних за допомогою агрегатів. Це визначення досить точно описує принципи роботи баз даних типу "ключ-значення", документ і сімейство значень. Агрегати полегшують роботу баз даних використовуючи кластери, оскільки агрегат являє собою природну одиницю реплікації і фрагментації. Крім того, агрегати спрощують розробку прикладних програм, які часто маніпулюють даними за допомогою агрегованих структур.

Бази даних типу "ключ-значення" і документні бази даних є сильно агрегатно-орієнтованими. Ці бази даних в основному були сконструйовані з агрегатів. Бази даних обох типів складаються з безлічі агрегатів, кожен з яких має ключ або ідентифікатор, який використовується для доступу до даних.

## Моделі NoSQL баз даних

1. Ключ-значення сховище використовує асоціативний масив як основну модель даних. Дані зберігаються без схеми і представляються як колекція пар типу «значення-ключ», при цьому ключ не може з'явитися в колекції більше одного разу. Це одна з найпростіших моделей, а багатші моделі зазвичай реалізовані як її розширення. Існують різні реалізації апаратного забезпечення, і деякі користувачі підтримують дані в пам'яті (RAM), в той час як інші використовують HDD накопичувачі або класичні обертові диски. В базі даних "ключ-значення" агрегат є непроникним для бази даних - просто великий чорний ящик, що складається переважно з бітів, які не несуть жодної інформації. Перевага непрозорості полягає в тому, що в агрегаті можна зберігати все що завгодно. База даних може обмежувати загальний розмір агрегату, але в іншому маємо повну свободу. Бази даних, що класифіковані як бази типу "ключ-значення", можуть пропонувати нові структури для даних, крім непрозорих агрегатів. Наприклад, база даних Riak дозволяє додавати метадані до агрегатів для індексування та встановлення зв'язків між агрегатами, а Redis дозволяє розбивати агрегати на списки або множини. Крім того, можна забезпечити механізм запитів за допомогою інтегрованих засобів пошуку, як в базі даних Solr. Наприклад, пошуковий механізм бази даних Riak, аналогічний пошуковому механізму бази Solr, виконує пошук агрегатів, що зберігаються у вигляді структур JSON або XML. У сховищі типу "ключ-значення" можна переглядати агрегат тільки за допомогою його ключа.

2. Документні бази – використовують концепцію ключ-значення але більш складну. Кожен документ в даному типі баз даних має свої дані та свій унікальний ключ. Цей тип орієнтований на документи, але дещо структурований. Документна база може бачити структуру агрегата. База даних накладає обмеження на те, що можна зберігати в агрегаті, визначаючи допустимі структури і типи. Однак це пдає можливість до гнучкого доступу. У документній базі даних можна посилати бази даних запити, що стосуються полів в агрегаті,

видаляти частину агрегату, а не весь агрегат, причому база даних може створювати індекси з урахуванням вмісту агрегату. На практиці розділова лінія між базами даних типу "ключ-значення" і документними базами даних дуже не чітка. Люди часто записують ідентифікатори в документні бази даних, щоб виконувати пошук в стилі "ключ-значення".

3. Сімейство колонок – ця модель зберігає таблиці даних як стовпчики, а не рядки. Це дозволяє забезпечити велику масштабованість та високу продуктивність.

Однією з ранніх і популярних баз даних NoSQL була база BigTable компанії Google [30]. Її ім'я викликає в уяві табличну структуру, що складається з окремих колонок і не має схеми. Однак цю структуру не слід представляти у вигляді таблиці; швидше вона являє собою дворівневий асоціативний масив. Ця модель вплинула на більш пізні бази даних, такі як HBase і Cassandra.

### **3.4.Порівняння SQL та NoSQL БД**

Не існує жодного підходу для всіх технологій баз даних. Саме тому використовують як реляційні, так і нереляційні бази даних для різних завдань. Навіть оскільки бази даних NoSQL посилюють популярність за їх швидкістю та масштабованістю, все ще є ситуації, коли доцільною є високо структурована SQL база даних.

#### **3.4.1.Переваги та недоліки SQL БД**

Переваги SQL бази даних:

1. Відповідність ACID (атомарність, узгодженість, ізолюваність, довговічність), що захищає цілісність бази даних, визначаючи точно, як транзакції взаємодіють з базою даних. Як правило, база даних NoSQL жертвує ACID для забезпечення гнучкості та швидкості обробки, але для електронної

комерції та фінансових додатків переважним варіантом залишається база даних, сумісна з ACID [31].

2. Кожна таблиця містить одну або декілька категорій даних у стовпцях[32].

3. Кожен рядок містить унікальний екземпляр даних для категорій, визначених стовпцями[33].

4. Користувач може отримати доступ до даних з бази даних, не знаючи структури таблиці бази даних[32].

Недоліки SQL бази даних:

1. Масштабованість: користувачі повинні масштабувати реляційні бази даних на потужних серверах, які є дорогими та складними для обробки. Щоб масштабувати реляційну базу даних, вона повинна бути розподілена на декілька серверів.[32]

2. Складність: дані SQL-сервера повинні бути вписані у таблиці. Якщо дані не вкладаються в таблиці, необхідно проектувати структуру бази даних, яка буде складною і важкою для обробки [32].

Вартість: РСКБД вимагають дорогих систем зберігання та запатентованих серверів [24].

### **3.4.2. Переваги та недоліки NoSQL БД**

Переваги NoSQL бази даних:

1. Зберігання великих обсягів даних, які погано структуровані. База даних NoSQL не обмежує типи даних, які можна зберігати разом, і дозволяє додавати нові типи. За допомогою баз даних на базі документів можна зберігати дані в одному місці, не визначаючи заздалегідь, які «типи» цих даних [34].

2. Швидша обробка даних, ніж в реляційних базах даних [32].

3. Дешевше: не реляційні бази даних використовують дешеві кластери товарних серверів для керування операціями та даними [33].

4. Масштабованість набагато краща, ніж в реляційних базах даних. Проте NoSQL бази даних не повністю масштабовані у всіх ситуаціях [35].

Недоліки NoSQL бази даних:

1. Узгодженість даних: більшість баз даних NoSQL не виконують транзакції ACID. Натомість NoSQL покладається на принцип "кінцевої послідовності". Це забезпечує деякі переваги продуктивності, але це створює ризик того, що дані на одному вузлі бази даних можуть не синхронізуватися з даними іншого вузла [35];

2. Відсутність стандартизації: NoSQL не є специфічним типом інтерфейсу бази даних або програмування. Мова дизайну та запитів баз даних NoSQL різко відрізняється між різними продуктами NoSQL - набагато ширше, ніж серед традиційних баз даних SQL [35].

Деякі не реляційні бази даних погано розповсюджуються на декілька вузлів. Якщо база даних не може відокремитись автоматично, вона не може автоматично збільшуватись або зменшуватись у відповідь на коливальний попит.

### **3.5. Система керування базами даних MS SQL**

Система керування базами даних (СКБД) — комплекс програмного забезпечення, що надає можливості створення, збереження, оновлення та пошуку інформації в базах даних з контролем доступу до даних.

MS SQL — вільна система керування реляційними базами даних. Вона використовується, в першу чергу, для створення динамічних веб-сторінок, оскільки має чудову підтримку з боку різноманітних мов програмування [36].

Гнучкість СКБД MS SQL забезпечується підтримкою великої кількості типів таблиць: користувачі можуть вибрати як таблиці типу ISAM, що підтримують повнотекстовий пошук, так і таблиці InnoDB, що підтримують транзакції на рівні окремих записів. Більш того, СКБД MS SQL поставляється із

спеціальним типом таблиць EXAMPLE, що демонструє принципи створення нових типів таблиць. Завдяки відкритій архітектурі і GPL-ліцензуванню, в СКБД MS SQL постійно з'являються нові типи таблиць.

Підтримуються операції відкриття та закриття БД з можливістю створення нової БД при відкритті та видалення існуючої при закритті. У термінах підсистеми "БД" системи OpenSCADA відкриттям БД є її реєстрація для подальшого використання у системі. Також підтримується операція запиту переліку таблиць у БД [34].

Основними тенденціями, які дали привід для проведення різних масштабних досліджень в області баз даних стали:

- Експонентний ріст даних. Обсяг даних, у тому числі синтетичних, що генеруються автоматизованими системами, значно зріс. Збільшилося і число прикладних областей, в яких вимагається обробка великих обсягів даних. До таких областей тепер відносяться не тільки традиційні корпоративні програми, пошук у веб, але також і наукові дослідження, обробка природних мов, аналіз соціальних мереж тощо.
- Значне ускладнення структур використовуваних даних. Прості види даних у вигляді чисел і символьних рядків стали доповнятися численною мультимедійною інформацією, просторовими, процедурними даними та великою кількістю інших складних форматів.
- Широке поширення дешевих високопродуктивних апаратних засобів. Щорічно спостерігаємо зростання обчислювальних можливостей мікропроцесорів, збільшення ємності і зниження вартості доступних і зручних в експлуатації пристроїв дискової і оперативної пам'яті.
- Активний розвиток засобів комунікації та «всесвітньої павутини» World Wide Web. WWW стає єдиним інформаційним середовищем, що пронизує весь світ і об'єднує величезне число користувачів та електронних пристроїв.

- Поява нових важливих областей застосування СКБД. У першу чергу, це пов'язано з інтелектуальним аналізом даних, сховищами даних, а останнім часом — з паралельними обчисленнями і хмарними технологіями.

Основними характеристиками СКБД:

- Контроль за надлишковістю даних
- Несуперечливість даних
- Підтримка цілісності бази даних (коректність та несуперечливість)
- Цілісність описується за допомогою обмежень
- Незалежність прикладних програм від даних
- Спільне використання даних
- Підвищений рівень безпеки

Можливості СКБД:

- Дозволяється створювати БД (здійснюється за допомогою мови визначення даних DDL (Data Definition Language))
- Дозволяється додавання, оновлення, видалення та читання інформації з БД (за допомогою мови маніпулювання даними DML, яку часто називають мовою запитів)
- Можна надавати контрольований доступ до БД за допомогою:
  1. Системи забезпечення захисту, яка запобігає несанкціонованому доступу до БД;
  2. Системи керування паралельною роботою прикладних програм, яка контролює процеси спільного доступу до БД;
  3. Система відновлення — дозволяє відновлювати БД до попереднього несуперечливого стану, що був порушений в результаті збою апаратного або програмного забезпечення.

### 3.6. Модель для MS SQL Server

Модель бази даних для MS SQL Server представлена на рис 3.2. та 3.3.

Важливі моменти:



- Первинні ключі представлені знаковими цілими числами, тому що в MS SQL Server немає можливості зробити bigint, int і smallint беззнаковими (tinyint, навпаки, буває тільки беззнаковим).
- Рядки представлені типом nvarchar довжиною до 150 символів (так і щоб гарантовано вкластися в обмеження 900 байт на довжину індексу;  $150 * 2 = 300$ , MS SQL Server для зберігання і порівняння символів в національних кодуваннях використовує два байта на символ).
- Поле sb\_is\_active представлено типом char довжиною в один символ (тому що в MS SQL Server немає типу даних enum), на це поле накладено обмеження check зі значенням [sb\_is\_active] IN ('Y', 'N').
- Поля sb\_start і sb\_finish представлені типом date (а не більше повними, наприклад, datetime), тому що ми зберігаємо дату видачі та повернення книги з точністю до дня.

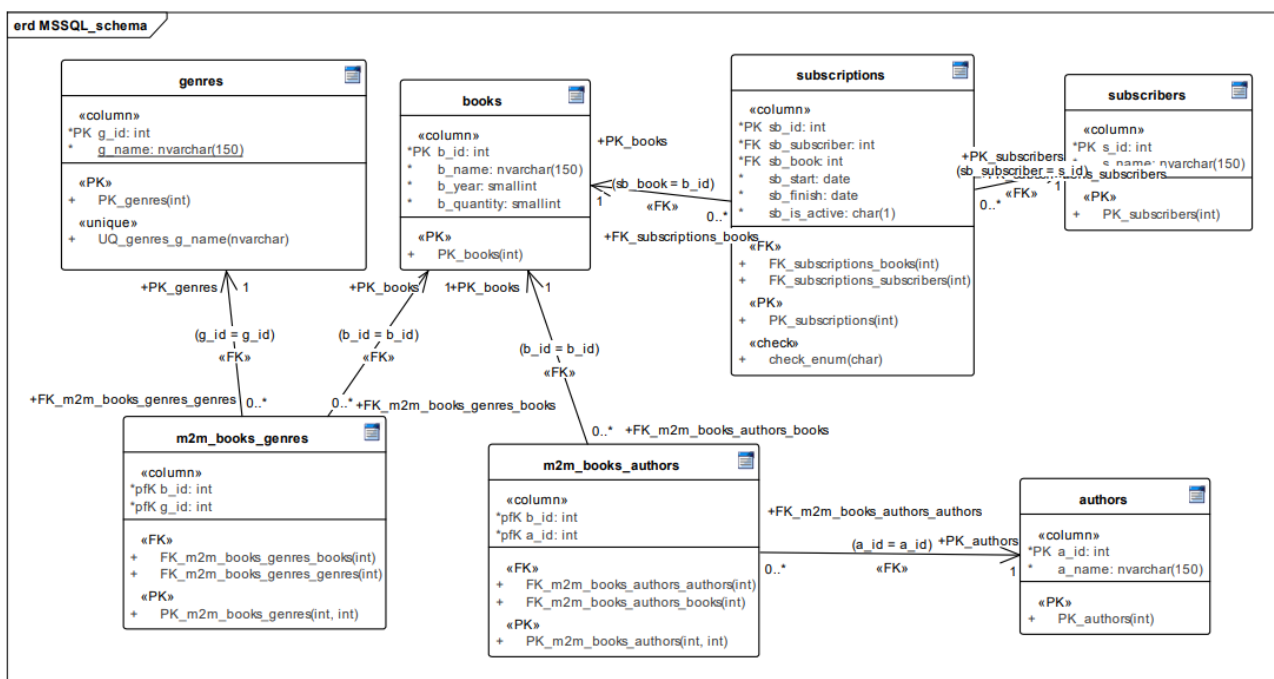


Рис 3.2 Модель бази даних для MS SQL Server в Sparx EA

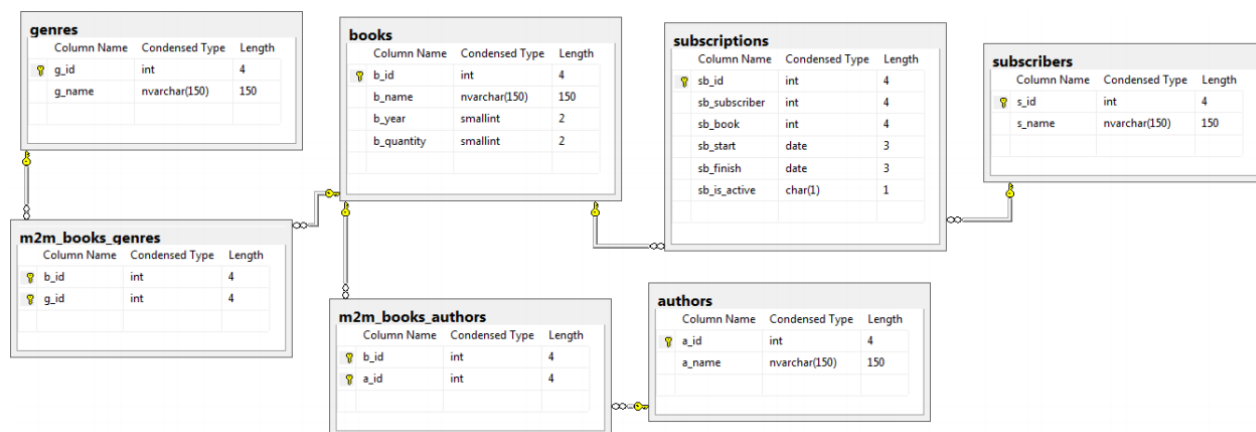


Рис 3.3 Модель бази даних для MS SQL Server в MS SQL Server Management Studio

Після формування та занесення даних в базу даних, таблиця бази даних системи буде виглядати, як показано в табл. 3.1.

Таблиця 3.1

Дата (день.місяць.рік)	Час	День тижня	Сезон	Напруга (В)	Струм (А)	Потужність (кВт)	Потужність за год. (кВт\год)
25.07.18	00:00	6	1	221	0.50	110.50	1052.34
25.07.18	00:30	6	1	220	0.57	125.40	1115.04
25.07.18	00:00	6	1	225	0.48	108.00	1169.04
25.07.18	01:30	6	1	210	0.49	102.90	1220.49
25.07.18	02:00	6	1	214	0.51	109.14	1275.06
25.07.18	02:30	6	1	220	0.50	110.00	1330.06
25.07.18	03:00	6	1	217	0.54	117.18	1388.65
...	...	...	...	...	...	...	...

Дані які були отримані в процесі використання пристрою збору та передачі інформації фіксуються в базі даних та зберігаються для подальшого аналізу.

### **Висновки до третього розділу**

На сьогоднішній день не існує єдиного підходу для всіх технологій баз даних. Саме тому в залежності від завдання використовують як реляційні, так і нереляційні бази даних. NoSQL бази даних стають все більш популярними за рахунок їх швидкодії та здатності до масштабування, але все одно є ситуації, коли перевага віддається більш структурованим SQL базам даних.

NoSQL бази даних, в порівнянні з SQL, набагато краще масштабуються, не накладають обмежень на типи даних, які можна зберігати разом, і дозволяють додавати нові типи при зміні потреб. З NoSQL базами даних можна зберігати різні типи даних в одному місці, не вказуючи завчасно які типи даних необхідно використовувати. Нереляційні бази даних дешевші і мають швидшу обробку даних. Але при цьому NoSQL бази даних жертвують відповідністю вимогам ACID для забезпечення гнучкості та швидкості обробки, що є не бажаним для фінансових додатків. Натомість нереляційні бази даних засновуються на принципі «кінцевої послідовності». Це забезпечує деякі переваги продуктивності, але створює ризик того, що дані на одному вузлі бази даних можуть не синхронізуватися з даними іншого вузла. Для збереження результатів вимірювань з датчиків, опрацювання великих даних, при розробці додатків для інтернету речей, веб-ресурсів та інших краще використовувати реляційні бази даних SQL бази даних.

Аналізуючи все, можна зробити висновок, що SQL бази даних необхідно використовувати, для забезпечення передачі інформації та надійності системи в зв'язках між таблицями. Якщо дані вдається структурувати, то далі працювати з ними набагато легше. Всі реляційні бази даних підтримують SQL, саме тому при зміні реляційних баз даних у користувача і розробника не виникне суттєвих проблем.

## РОЗДІЛ 4. ПРИСТРІЙ ЗБОРУ, ОБРОБКИ ТА ПЕРЕДАЧІ ДАНИХ

Для забезпечення прийому даних та передачі їх до бази даних потрібно використати модулі фіксації та підрахунку електричної енергії та модулі, які забезпечують зв'язок та передачу даних до бази даних.

### 4.1. Пристрої фіксації електричної енергії

#### 4.1.1 Ватметр TS-836

У кожному будинку встановлюється лічильник електроенергії, що підраховує і відображає кількість використаної електроенергії в конкретній квартирі або будинку. Щоб дізнатися споживану потужність конкретного пристрою, можна скористатися і тестером, зробивши виміри споживаного струму. Даний пристрій який зображений на рис. 4.1. підходить для підрахунку та відображення необхідної інформації.



Рис 4.1 Ватметр TS-836

Пристрій має такі характеристики:

- робоча напруга: 230 ~ 250В / 50Гц;
- вимірювана потужність: 1 ~ 3600Вт;
- враховується споживання: 0.0001 ~ 999.9 кВт год;
- максимальна сила струму: 16 а;

- враховується вартість: 0 ~ 9,999 грошових одиниць;
- точність вимірювання:  $\pm 1\%$ ;
- умови експлуатації: Температура:  $-10 \sim 60^{\circ} \text{C}$ ; Вологість: 10 ~ 95%.

Функції:

- вимір напруги;
- вимір струму;
- вимір частоти електромережі;
- вимір витрати електроенергії;
- вимір споживаної потужності;
- моніторинг активного навантаження.

Також є можливість встановлення вартості електроенергії, що дуже зручно для розрахунків економії електричної енергії. Але не має інтерфайсу для отримання даних для передачі та подальшої їх обробки.

Ватметр в реальному часі аналізує роботу підключеного через нього пристрою. Пристрій має свою номінальну потужність, зазначену заводом виробником. За допомогою функції вимірювання потужності можна визначити робочу потужність цих приладів і зробити висновок, на скільки ефективно працюють пристрої. Пристрій має свою батарею і зберігає всі свідчення, навіть якщо не підключено в розетку. Є кнопка RESET, яка скидає всі значення, для вимірювання нових.

#### **4.1.2. «Розумна розетка» BroadLink SP3S**

«Розумна розетка» фірми BroadLink модель SP3S Contros з вбудованим ватметром (рис. 4.2).

Відмінною особливістю даного пристрою є вбудований лічильник електроспоживання, який дозволить контролювати і зберігати показники витрат електроенергії підключеного пристрою.



Рис 4.2 «Розумна розетка» фірми BroadLink модель SP3S Contros

Пристрій створений для полегшення контролю за електроприладами і економії електроенергії. Керування здійснюється за допомогою додатку, встановленого на смартфон на платформі IOS і Android. Розетка SP3S дозволяє регулювати активність домашніх приладів у встановлений час у відповідності до сценаріїв наявності або відсутності господаря на території будинку. Якщо електричний прилад довгий час знаходиться в режимі очікування, розумна розетка відключає його в цілях безпеки та економії енергії. Вогнетривкий матеріал і захист від перенапруги і короткого замикання забезпечить надійну і безпечну експлуатацію розетки SP3S Contros. Пристрій ефективно працює в комплекті з іншими пристроями розумного будинку, такими як система охорони SmartOne і E-air.

основними характеристиками пристрою є:

- температура роботи від 0 - 50 c;
- робоча вологість менше 80%;
- бездротова потужність - <0.5 вт;
- максимальне навантаження - 16а / 3200w
- напруга - ac100 ~ 240v ac 50 / 60hz;
- частота Wi-Fi-сигналу - 2,4 ггц 802.11b / g / n.;
- вбудований лічильник електроспоживання.

### 4.1.3. Монітор електроенергії PZEM-004T

Модуль призначений для аналізу та моніторингу споживання електроенергії, легко стикується з платами Arduino, процесорами, системами "розумний будинок". Пристрій зображений на рис. 4.3.



Рис 4.3 Монітор електроенергії PZEM-004T

Модуль має гальванічески розв'язаний послідовний опто-інтерфейс. Основний процесор модуля - мікросхема SD3004 від китайської компанії SDIC Microelectronics Co., Ltd призначена для вимірювання електроенергії. Крім того на платі є мікросхема EEPROM Atmel 24C02C і дві оптопари PC817, що забезпечують гальванічну розв'язку послідовного інтерфейсу. Вимірювання напруги проводиться безпосередньо дільником, а вимір струму - через струмовий трансформатор. Пристрій має функцію, що дозволяє налаштувати поріг потужності, при досягненні якого буде спрацьовувати зумер. У разі раптового відключення електроживлення вже виміряні параметри будуть збережені. Діапазон робочих напруг досить широкий: 80-260 В. Одна з істотних особливостей даного модуля - це можливість її підключення до комп'ютера за допомогою інтерфейсу Uart.

Характеристики:

- вимірювання поточного напруга мережі змінного струму 80-260в ;
- вимірювання струму 0 - 100а;
- вимірювання активної потужності 0 - 22квт ;
- вимірювання спожитої електроенергії;

- точність вимірювання 1%;
- робоча частота 45-65Гц;
- послідовний інтерфейс uart зі швидкістю 9600;
- напруга живлення 5в;
- можливість підключення lcd або led дисплеїв;
- розмір плати 31х74мм;
- діаметр котушки трансформатора струму 33мм;

#### Переваги

- прилад відмінно працює з великими струмами.
- відрізняється високою точністю вимірювань.
- якісно виконана конструкція.
- може бути використаний в якості лічильника електроенергії або в якості дистанційного монітора електроспоживання.

Для якості демонстраційного пристрою було обрано монітор електроенергії PZEM-004T. Написано програмне забезпечення у середі Arduino IDE та протестована. Скріншоти веб-інтерфейсу зображено на рис 4.4.

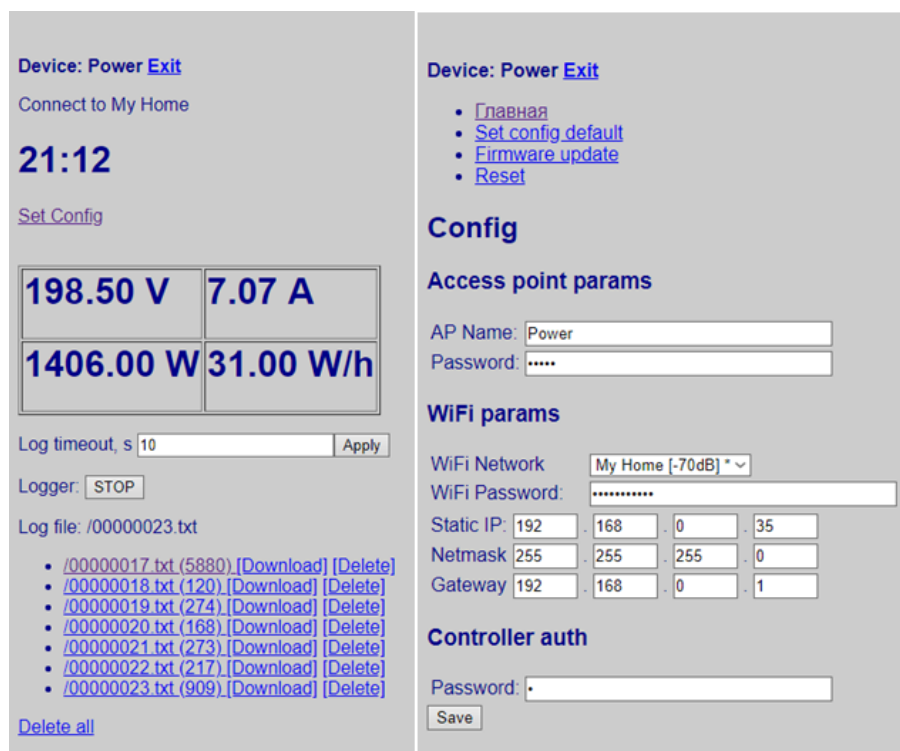


Рис 4.4 Скріншот веб-інтерфейсу модуля PZEM-004T



При першому включенні, пристрій вмикає свій wi-fi доступ. Після підключення необхідно на сторінці налаштувань задати локальну wi-fi мережу та надати статичну IP адресу та перезапустити пристрій.

Пристрій підключається до локальної мережі на базі даних для передачі параметрів. Фіксує напругу мережі, струм протікаючі через навантаження, обраховує потужність та фіксує витрачену потужність на годину. Упаковка даних обробляється контролером та передається через wi-fi мережу до бази даних. При відсутності зв'язку з базою даних пристрій зберігає дані у своїй внутрішній пам'яті та після з'єднання відправляє до бази даних. Передбачена функція вибору частоти зберігання даних.

## **4.2. Open Data Protocol**

Для передачі, запису та отримання даних від бази даних необхідно використовувати протоколи транспортування даних. Для великих систем потрібен надійний та відмово-стійкий протокол передачі даних до бази даних який би підтримував HTTP-команди які використовує пристрій збору та передачі даних. Одним з таких протоколів є OData протокол.

Open Data Protocol (OData) - це відкритий веб-протокол для запиту і поновлення даних. Протокол дозволяє виконувати операції з ресурсами, використовуючи в якості запитів HTTP-команди, і обмінюватися даними в форматах JSON або XML. OData є одним з кращих стандартів для створення RESTful API.

Базові можливості OData

- Просте читання даних (запити без параметрів);
- Отримання колекції об'єктів;
- Отримання окремого об'єкта;
- Отримання окремої властивості;
- Отримання об'єктів зі зв'язків;

- Параметризовані запити (пошук, сортування, і т.д.);
- Пошук по критеріям ( "менше", "більше" і т.п.), можливість побудови складних умов, використовуючи логічні вирази;
- Пошук по зв'язках (по зв'язаних об'єктів);
- Сортування по будь-якому набору полів;
- Отримання складових складних об'єктів, наприклад, отримати дані про людину разом з його телефонами і списком друзів;
- Додавання даних;
- Додавання простих об'єктів;
- Додавання колекцій об'єктів;
- Додавання складових об'єктів (разом з вкладеними об'єктами);
- Модифікація даних;
- Модифікація окремих властивостей (полів);
- Заміна об'єкта цілком на новий;
- Видалення даних.

#### Бібліотеки для роботи з OData

В даний час існує велика кількість бібліотек, що підтримують протокол OData, і кожен день з'являються нові.

Зокрема, з OData працюють такі JavaScript бібліотеки, як Kendo UI, DevExtreme Web, Syncfusion HTML5 controls, Infragistics HTML5 controls, OpenUI5, Wijmo, JayData, Breeze.js, datajs, ODataJS, angular-odata, і т.д. Існують також бібліотеки для Java, .NET, C ++, Python, Objective-C і т.д.

Багато з цих бібліотек істотно спрощують розробку великих програм завдяки стандарту OData. Наприклад, необхідність вказати URL до сервера OData і все інше він зробить сам: пейджінг, сортування, додавання-модифікацію-видалення записів, фільтрацію даних, угруповання і т.д.

### **Висновки до четвертого розділу**

Системи фіксації, обробки та передачі інформації в MicroGrid являють собою інформаційно - керуючі системи, що забезпечують координоване отримання даних від споживачів та генераторів електричної енергії, за допомогою засобів та конфігурацій мережі. Розроблено тестовий макет пристрою збору та передачі даних.

При моделюванні та збору пристрою збору даних було використано модуль призначений для аналізу та моніторингу споживання електроенергії «PZEM-004T». Було написано програмне забезпечення в середовищі розробника Arduino IDE та протестовано на різних пристроях.

Описан протокол передачі даних «Open Data Protocol» від пристрою збору даних та базою даних. Наведені переваги та недоліки використання даного протоколу передачі.

## РОЗДІЛ 5. РОЗРОБКА СТАРТАП – ПРОЕКТУ

Стартап як форма малого ризикового (венчурного) підприємництва впродовж останнього десятиліття набула широкого розповсюдження у світі через зниження бар'єрів входу в ринок (із появою Інтернету як інструменту комунікацій та збуту стало простіше знаходити споживачів та інвесторів, займатись пошуком ресурсів, перетинати кордони між ринками різних країн), і вважається однією із наріжних складових інноваційної економіки, оскільки за рахунок мобільності, гнучкості та великої кількості стартап-проектів загальна маса інноваційних ідей зростає.

Проте створення та ринкове впровадження стартап-проектів відзначається підвищеною мірою ризику, ринково успішними стає лише невелика частка, що за різними оцінками складає від 10% до 20%. Ідея стартап-проекту, взята окремо, не вартує майже нічого: головним завданням керівника проекту на початковому етапі його існування є перетворення ідеї проекту у працюючу бізнес-модель, що починається із формування концепції товару (послуги) для визначеної клієнтської групи за наявних ринкових умов.

### **Етапи розроблення стартап-проекту:**

#### **1. Маркетинговий аналіз стартап-проекту**

В межах цього етапу:

- розробляється опис самої ідеї проекту та визначаються загальні напрями використання потенційного товару чи послуги, а також їх відмінність від конкурентів;
- аналізуються ринкові можливості щодо його реалізації;
- на базі аналізу ринкового середовища розробляється стратегія ринкового впровадження потенційного товару в межах проекту.

#### **2. Організація стартап-проекту**

В межах цього етапу:

- складається календарний план-графік реалізації стартап-проекту;

- розраховується потреба в основних засобах та нематеріальних активах;
- визначається плановий обсяг виробництва потенційного товару, на основі чого формулюється потреба у матеріальних ресурсах та персоналі;
- розраховуються загальні початкові витрати на запуск проекту та планові загальногосподарські витрати, необхідні для реалізації проекту.

### 3. Фінансово-економічний аналіз та оцінка ризиків проекту

В межах цього етапу:

- визначається обсяг інвестиційних витрат;
- розраховуються основні фінансово-економічні показники проекту (обсяг виробництва продукції, собівартість виробництва, ціна реалізації, податкове навантаження та чистий прибуток) та визначаються показники інвестиційної привабливості проекту (запас фінансової міцності, рентабельність продажів та інвестицій, період окупності проекту);
- визначається рівень ризикованості проекту, визначаються основні ризики проекту та шляхи їх запобігання (реагування на ризики).

### 4. Заходи з комерціалізації проекту

Цей етап спрямовано на пошук інвесторів та просування інвестиційної пропозиції (оферти). Він передбачає:

- визначення цільової групи інвесторів та опису їх ділових інтересів;
- складання інвест-пропозиції (оферти): стислої характеристики проекту для попереднього ознайомлення інвестора із проектом;
- планування заходів з просування оферти: визначення комунікаційних каналів та площадок та планування системи заходів з просування в межах обраних каналів;
- планування ресурсів для реалізації заходів з просування оферти.

Означені етапи, реалізовані послідовно та вчасно – створюють передумови для успішного ринкового старту [62].

В рамках магістерської дисертації буде розглянуто перший етап розробки стартап-проекту, а саме маркетинговий аналіз стартап-проекту системи

прогнозування залишку електроенергії, виробленої альтернативними джерелами MicroGrid для подальшої реалізації на локальному ринку електроенергії.

### 5.1. Опис ідеї проекту

У табл.5.1 наведено опис ідеї стартап-проекту, напрямки застосування та переваги.

Таблиця 5.1.

Опис ідеї стартап - проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
<u>Зміст ідеї:</u> створення системи збору, обробки та прогнозування кількості залишкової енергії на майбутній період за допомогою кореляційно-регресійного аналізу даних про генерацію та споживання у MicroGrid з альтернативними та відновлювальними джерелами енергії	Системи розподіленої генерації MicroGrid	Можливість контролю за витраченою та генерованою електричної енергії; економічна вигода від продажу надлишку енергії, виробленої альтернативними джерелами, за «зеленим тарифом»

У табл.5.2 наведено характеристики ідеї проекту та визначені сильні, слабкі та нейтральні характеристики ідеї проекту.

Таблиця 5.2.

№ п/ п	Техніко- економі чні характер истики ідеї	(потенційні) товари/концепції конкурентів				W (слаб ка сторо на)	N (нейт ральн а сторо на)	S (силь на стор она)
		Мій проект	Система моніто- рингу JEGRID	Система AMIGO	Artificial Neural Networks			
1.	Система збору даних	Отриман ня чотирьох параметрі в енергосп оживання навантаж енням	Отрима ння двох парамет рів енергос поживан ня наванта женням	Отрима ння двох парамет рів енергос поживан ня наванта женням без відправк и на сервер	Отриман ня двох парамет рів енергосп оживанн я наванта женням без можливо сті зберіган ня		+	
2.	Метод підраху нку залишко вої енергії	Лінійна регресія	Не ведеться	Не ведеться	Не ведеться			+
3.	Інтерфе йс програм ного забезпеч ення	У динаміці та статиці	У статиці	У статиці	У статиці			+

## 5.2. Технологічний аудит ідеї проекту

У табл.5.3 наведена технологічна здійсненність ідеї проекту.

Таблиця 5.3.

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Отримання даних використання та генерації електричної енергії	Розробка, дослідження, програмування;	Наявна	Доступна
2	Метод аналізу та прогнозування залишкової енергії на майбутній період	Створення програмного забезпечення для прогнозування;	Не наявна	Доступна
Обрана технологія реалізації ідеї проекту: самостійна розробка на основі дослідження				

### 5.3. Аналіз ринкових можливостей запуску стартап-проекту

У табл.5.4 наведена попередня характеристика потенційного ринку стартап проекту.

Таблиця 5.4.

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	100
2	Загальний обсяг продаж, грн/ум.од	700
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Відсутність масового виробника
5	Специфічні вимоги до стандартизації та сертифікації	Немає
6	Середня норма рентабельності в галузі (або по ринку), %	31

У табл.5.5 наведена характеристика потенційних клієнтів стартап-проекту.



Таблиця 5.5. Характеристика потенційних клієнтів стартап-проекту

Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
Можливість контролю споживання та генерації електричної енергії. Точний прогноз використання та генерації електричної енергії Потреба у якнайшвидшій окупності встановлених систем альтернативної генерації - економічна вигода від продажу надлишку виробленої енергії	Приватні будинки; Системи MicroGrid та Smart House; Підприємства, заводи та великі організації;	Експлуатація з різними пристроями навантаження та генерації електричної енергії;	- до продукції: Точність; Надійність; Дешевизна; Якість; - до компанії-постачальника: Точність; Брендинг та відомість; Гарантійність;

У табл.5.6 наведений факторний загроз.

Таблиця 5.6.

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Захищеність інформації	Складність персоналізації інформації;  Демпінг;	Зробити упор на надійність та розробити допоміжні системи захисту персональної інформації та її передачі;
2	Конкуренція		
3	Зберігання інформації		

У табл.5.7 наведені фактори поживностей.

Таблиця 5.7.

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Достовірність і надійність інформації	Переваги при передачі інформації до бази даних	Зроблений упор на маркетинг та рекламу у цих напрямках для рекомендування себе, як компанії, на ринку;
2	Безпомилковість		

У табл.5.8 наведений ступеневий аналіз конкуренції на ринку.

Таблиця 5.8.

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Тип конкуренції: чиста	В кого краще - в того купують	Покращення товару та сфери обслуговування
2. За рівнем конкурентної боротьби: локальна	Належить до медичного ринку збуту;	Розширення функціоналу та орієнтації користувачів

3. За галузевою ознакою: міжгалузева	Притаманна різним галузям застосування;	Розширення функціоналу та галузей застосування
4. Конкуренція за видами товарів: товарно-родова та товарно-видова	Належить до систем Machine Learning	Розширення функціоналу пристрою
5. За характером конкурентних переваг: цінова та нецінова	Чим дешевше – тим привабливіше; Чим краще – тим рентабельніше;	Покращення цінової політики та якості товару
6. За інтенсивністю: не марочна	Не жорстка конкуренція	Агресивні та не агресивні форми піару

У табл.5.9 наведений SWOT- аналіз стартап-проекту.

Таблиця 5.9.

Сильні сторони: Незалежна система збору даних; Цілодобовий аналіз зібраних даних; Прогноз з малою похибкою; Надійні протоколи передачі даних.	Слабкі сторони: Не захищені дані;
Можливості: Отримання прогнозу на майбутній період; Отримання інформації про споживання та генерацію електричної енергії як локально так і в мережі інтернет;	Загрози: Вихід з ладу пристрою збору інформації; Втрата даних; Захист інформації;

#### 5.4. Розроблення маркетингової програми стартап-проекту

У табл.5.10 наведена базова стратегія конкурентної поведінки.

Таблиця 5.10.

№ п/п	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки*
1	З погляду систем збору інформації про споживання те генерацію енергії – ні, є багато систем конкурентів які вже здатні на це, а з погляду системи яка має можливість аналізувати результати та одержувати прогноз на майбутній період – так.	Буде шукати нових розширяючі функціонал і потенціал продукту, а також існуючі клієнти у конкурентів самовільно будуть використовувати більш кращий продукт	Ні, не буде, так як це зменшить клієнтську базу	Помірна, місцями агресивна

У табл.5.11 наведені ключові переваги концепції потенційного товару.

Таблиця 5.11.

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	«Ефективність понад усе»; «Бути першим»; Швидка та точна оцінка; Прогнозування для Вашої економії;	Отримання точного прогнозу на найближчий період часу	Створення надійного бренду; Постійний розвиток та апгрейд системи та компанії у всіх напрямках; Заняття усіх можливих ніш у енергетичній галузі для створення монополії; Дотримуватися схеми ціна – якість;

У табл.5.12 наведені межі встановлення ціни на товар.

Таблиця 5.12.

№ п/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
	130-150% від ціни нашого продукту	150-170% від ціни нашого продукту	20000 - 100000 грн зі 100 проданих од.	1500/4000 грн

У табл.5.13 наведено принцип формування системи збуту.

Таблиця 5.13.

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Задоволення потреб, спрощення деяких функцій користувачів у своїй роботі,	Збут товару та задоволення запитуваних потреб клієнтів	Усі можливі канали збуту (глибока)	Власна
2	отримання точного прогнозування за лічені хвилини,  Залучення до новітніх інноваційних технологій  Участь у програмах енергозбереження, захисту навколишнього середовища, «зеленої» енергетики	Збут та реклама товару та задоволення запитуваних потреб клієнтів	Усі можливі канали збуту (глибока)	Залучена

### **Висновки до п'ятого розділу**

Згідно проведеного аналізу розроблюваний проект має можливість ринкової комерціалізації. Зростання попиту на аналогічні товари додає масовості придбання подібних пристроїв та систем, але створює жорсткі конкурентні умови виходу на ринок, де динаміка ринку доволі сприятлива до проекту.

Проект має високі перспективи впровадження з огляду на потенційні групи клієнтів, якими виступають великі підприємства, приватні будинки з системами MicroGrid та Smart House та організації з використанням відновлювальних джерел живлення. Бар'єрами входження на ринок можуть бути відсутність масового виробника та сильний конкурентний тиск з боку великих фірм аналогічних продуктів. Але якщо вести помірно агресивну боротьбу в конкурентному середовищі, проект має великі шанси та можливість зарекомендувати бренд, де в подальшому здобудеться місце на ринковій економіці.

Подальша імплементація проекту є доцільною та рентабельною.

## ВИСНОВКИ

В ході роботи було розглянуто різні методи прогнозування оцінки ефективного використання електричної енергії в MicroGrid та детально описан метод регресійного аналізу як спосіб швидкого та з мінімальною похибкою отримання оцінки ефективності використання системи та для заощадження на продажі залишкової електричної енергії в мережу.

В роботі було показано, що проведений кореляційно-регресійний аналіз підтверджує існування статистичного взаємозв'язку між прибутком від реалізації електроенергії за зеленим тарифом, цінами на зелений тариф та цінами на електроенергію в мережі. Обчислення коефіцієнтів двофакторної регресійної моделі встановлює рівень залежності між параметрами та дозволяє прогнозувати економічні параметри локального ринку MicroGrid для планування обсяги закупівель та продажів електроенергії його учасниками.

Було написано та детально описано структуру демонстраційної програми для знаходження оцінки кількості залишкової електроенергії за допомогою якої в наступному місяці можна заощадити.

В роботі було розглянуто можливі бази даних для збереження даних та подальшого їх використання. Розглянуто та описано переваги та недоліки баз даних та обрано більш гнучку та відмовостійку базу для формування таблиць даних.

В ході роботи було розглянуто пристрої збору та передачі даних, виявлено недоліки та переваги та обрано єдину платформу для прийому, обробки та передачі даних в базу даних. Описано принцип та протоколи передачі даних.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Жуйков В. Я., Ямненко Ю. С., Бойко І. Ю., Клепач Л. Є. Статична та динамічна тарифікація електроенергії автономних Micro Grid. Вісник ЖДТУ, №3 (78). – 2016 – С. 12–17.
2. Ледин С.С. Интеллектуальные сети SmartGrid – будущее российской энергетики (ЗАО ИТФ «Системы и технологии») // Автоматизация и IT в энергетике. – 2010. – № 11(16). – С. 4–8.
3. Thirugnanam, K.; Kerk, S.K.; Yuen, C.; Liu, N.; Zhang, M. Energy Management for Renewable Micro-Grid in Reducing Diesel Generators Usage with Multiple Types of Battery. IEEE Trans. Ind. Electron. 2018 – P. 24–25.
4. Кириленко О.В., Блінов І.В., Попович В.І., Олефір Д.О., Методологія об'єктно-орієнтованого моделювання для опису функціонування конкурентного оптового ринку електричної енергії. Проблеми загальної енергетики. 2011. Вип. 1(24). 5-10 с.
5. Харченко М.А. Корреляционный анализ: Учебное пособие для вузов. – Воронеж: Изд-во ВГУ, 2008. – 31 с.
6. Елтышев Д.К., Хорошев Н.И. Системный подход к формированию и реализации программ энергосбережения и повышения энергетической эффективности // Фундаментальные исследования. – 2014. – № 5–4. – С. 697–701.
7. Франк Т., Кычкин А.В., Мусихина К.Г. Государственное управление проектами в области энергосбережения как база для эффективного внедрения лучших практик // Менеджмент в России и за рубежом. – 2014. – № 3. – С. 98–104.
8. Morariu N., Iancu E., Vlad S. A neural network model for time series forecasting // Romanian Journal of Economic Forecasting. 2009, No. 4. P. 213 - 223.



9. Mazengia D.H. Forecasting Spot Electricity Market Prices Using Time Series Models: Thesis for the degree of Master of Science in Electric Power Engineering. Gothenburg, Chalmers University of Technology, 2008. 89 p – P. 20.

10. Петроченков А.Б., Франк Т., Ромодин А.В., Кычкин А.В. Полунатурное моделирование активно-адаптивной электрической сети // Электротехника. – 2013. – № 11. – С. 60–63.

11. Елтышев Д.К., Петроченков А.Б., Бочкарёв С.В. К вопросу о применении генетических методов для решения задач поддержки жизненного цикла электрооборудования // Доклады Томского государственного университета систем управления и радиоэлектроники. – 2009. – Т. 2. – С. 136–142.

12. Некоторые аспекты технологии эксплуатации электротехнических объектов на основе методов краткосрочного прогнозирования технического состояния / В.П. Казанцев, А.Б. Петроченков, А.В. Ромодин, Н.И. Хорошев // Электротехника. – 2011. – № 11. – С. 28–34.

13. Кычкин А.В. Долгосрочный энергомониторинг на базе программной платформы OPENJEVIS // Вестник Пермского национального исследовательского политехнического университета. Электротехника, информационные технологии, системы управления. – 2014. – № 1 (9). – С. 5–15.

14. Хорошев Н.И., Елтышев Д.К., Кычкин А.В. Комплексная оценка эффективности технического обеспечения энергомониторинга // Фундаментальные исследования. – 2014. – № 5–4. – С. 716–720.

15. Новиков В.В. Интеллектуальные измерения на службе энергосбережения // Энергоэксперт. – 2011. – № 3. – С. 15–20.

16. Шишкин П.П. Кычкин А.В. «Фундаментальные исследования. – 2014. – № 11 (часть 7) » – С. 1536-1540.

17. Jingfei Yang M. Sc. Power System Short-term Load Forecasting: Thesis for Ph.d degree. Germany, Darmstadt, Elektrotechnik und Informationstechnik der Technischen Universitat, 2006. 139 p.

18. Методи прогнозування. [Електронний ресурс] - Режим доступу до ресурсу: <http://wiki.tntu.edu.ua/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4%D0%B8%D0%BF%D1%80%D0%BE%D0%B3%D0%BD%D0%BE%D0%B7%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F#.D0.9C.D0.B5.D1.82.D0.BE.D0.B4.D2.D0.93.D1.83.D1.81.D0.B5.D0.BD.D0.B8.D1.86.D1.96.22.SSA>

19. Пальчиков, А.С. Существующие способы прогнозирования электропотребления объектов в металлургической отрасли //Современные научные исследования и инновации. 2012. № 9. С. 12-17.

20. Prajakta S.K. Time series Forecasting using Holt-Winters Exponential Smoothing // Kanwal Rekhi School of Information Technology Journal 2004. 13 p.

21. Jingfei Yang M. Sc. Power System Short-term Load Forecasting: Thesis for Ph.d degree. Germany, Darmstadt, Elektrotechnik und Informationstechnik der Technischen Universitat, 2006. 139 p.

22. Методи прогнозування. [Електронний ресурс] - Режим доступу до ресурсу: <http://wiki.tntu.edu.ua/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4%D0%B8%D0%BF%D1%80%D0%BE%D0%B3%D0%BD%D0%BE%D0%B7%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F#.D0.9C.D0.B5.D1.82.D0.BE.D0.B4.D2.D0.93.D1.83.D1.81.D0.B5.D0.BD.D0.B8.D1.86.D1.96.22.SSA>

23. Prajakta S.K. Time series Forecasting using Holt-Winters Exponential Smoothing // Kanwal Rekhi School of Information Technology Journal 2004. 13 p.

24. Тихонов Э.Е. Прогнозирование в условиях рынка. Невинномысск, 2006. 221 с.

25. Gheyas I.A., Smith L.S. A Neural Network Approach to Time Series Forecasting // Proceedings of the World Congress on Engineering, London, 2009, Vol 2 P. 1292 - 1296.

26. Pradhan R.P., Kumar R. Forecasting Exchange Rate in India: An Application of Artificial Neural Network Model // Journal of Mathematics Research. 2010, Vol. 2, No. 4. P. 111 - 117.

27. Zhang Y. Prediction of Financial Time Series with Hidden Markov Models: Master of applied science: Simon fraser university / School of Computer Science. — 2005. — P. 102.

28. International Food Policy Research Institute 1999. 59 p.

29. MySQL 5.6 Reference Manual [Электронный ресурс] – Режим доступа до ресурсу: <https://dev.mysql.com/doc/refman/5.6/en/>.

30. David Hows, Peter Membrey, Eelco Plugge, Tim Hawkins. The Definitive Guide to MongoDB: A complete guide to dealing with Big Data using MongoDB, Third Edition. — Apress, 2015. — 376 с.

31. Callaghan T. Sysbench Benchmark for MongoDB [Электронный ресурс] / Tim Callaghan // Tokutek, TokuView – Режим доступа до ресурсу: [www.percona.com/blog/2013/03/14/sysbench-benchmark-for-mongodb/](http://www.percona.com/blog/2013/03/14/sysbench-benchmark-for-mongodb/).

32. Richards J. Advantages of NoSQL Databases [Электронный ресурс]/Jenny Richards – Режим доступа до ресурсу: [bigdata-madesimple.com/advantages-of-nosql-databases/](http://bigdata-madesimple.com/advantages-of-nosql-databases/).

33. The Limitations of NoSQL Database Storage [Электронный ресурс] – Режим доступа до ресурсу: <http://www.channelfutures.com/cloud-services/limitations-nosql-database-storage-why-nosqls-not-perfect>.

34. Wodehouse C. SQL vs. NoSQL Databases: What's the Difference? [Электронный ресурс] / Carey Wodehouse – Режим доступа до ресурсу: [www.upwork.com/hiring/data/sql-vs-nosql-databases-whats-the-difference/](http://www.upwork.com/hiring/data/sql-vs-nosql-databases-whats-the-difference/).

35. 10gen — ещё один стартап в области Cloud Computing [Электронный ресурс] – Режим доступа до ресурсу: [special.habrahabr.ru/kyocera/p/3660/](http://special.habrahabr.ru/kyocera/p/3660/).

36. Eelco Plugge, Peter Membrey, Tim Hawkins. The Definitive Guide to MongoDB: The NoSQL Database for Cloud and Desktop Computing. — Apress, 2010. — 327 с.

37. Закон України Про внесення змін до деяких законів України щодо забезпечення конкурентних умов виробництва електроенергії з альтернативних джерел енергії. URL: <http://zakon3.rada.gov.ua/laws/show/514-19> (дата звернення 19.09.2018)

38. Военно-экономический анализ: Учеб. / под общ. ред. С.Ф. Викулова. – Ярославль: ВФЭИ, 2010 – С. 10.

39. Интеллектуальная энергетическая система России: проект реализации технологической платформы // Российское энергетическое агентство. – М., 2010. – 169 с.

40. Sustainable Energy Management Solution - GridLogic | SolarCity. URL: <http://www.solarcity.com/commercial/sustainable-energy-solution>.

41. Conejo A. J., Morales J. M., Baringo L. Real-time demand response model, IEEE Trans. on Smart Grid. 2010, 236–242.

42. Ferreira R. S., Barroso L. A. N., Carvalho M. M. Demand response models with correlated price data: a robust optimization approach, App. Energy, 96, 2012, 133–149.

43. Lujano-Rojas J. M. Optimum residential load management strategy for real time pricing demand response programs, Energy policy, 45, 2012, 671–679.

44. Khalili A. H., Wu C., Aghajan H. Hierarchical preference learning for light control from user feedback, IEEE Computer Society Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW 2010), 2010, 56–62.

45. Gottwalt S. Demand side management — a simulation of household behavior under variable prices, Energy policy, 2011, 8163–8174.

46. Li B. Predicting user comfort level using machine learning for smart grid environments, Innovative Smart Grid Technologies (ISGT 2011), 1–6.

47. Trevor Hastie, Robert Tibshirani, and Jerome Friedman, The Elements of Statistical Learning, Springer New York Inc., USA, 2009, P. 10.

48. Rob Hyndman and George Athanasopoulos, Forecasting: principles and practice. OTexts: Melbourne, Australia, 2013. URL: <http://otexts.org/fpp/>.

49. Adam Zagdański, Artur Suchwałko, Analiza i prognozowanie szeregów czasowych. Praktyczne wprowadzenie na podstawie środowiska R (in Polish). PWN, 2015, 213p.

50. Zhuikov V. Y., Tereschenko T. O., Petergerya, J.S Preobrazovanie diskretnyih signalov na konechnyih intervalah v orientirovannom bazise – K.:Avers, 2004. – 274 p.

51. Харченко М.А. Корреляционный анализ: Учебное пособие для вузов. – Воронеж: Изд-во ВГУ, 2008. – 31 с.

52. Дрейпер Н., Смит Г. Прикладной регрессионный анализ. М.: Издательский дом «Вильямс». 2007.

53. Айвазян С.А. Прикладная статистика и основы эконометрики. М.: Юнити. 2001– 21 с.

54. Брандт З. Анализ данных. М.: Мир. 2003 – 16 с.

55. Стрижов В. В. Методы индуктивного порождения регрессионных моделей. М.: ВЦ РАН. 2008. 55 с.

56. Стрижов В.В., Крымова Е.А. Методы выбора регрессионных моделей. М.: ВЦ РАН, 2010. 60 с.

## ABSTRACT

**Actuality of theme.** Current research aimed at increasing the efficiency of the use of electric energy in distributed generation MicroGrid systems and redistribution of energy flows in local virtual markets emphasize the relevance of monitoring, estimating and predicting energy generation targets. From the point of view of joint consideration of technical and economic factors of MicroGrid functioning, demand level, electricity price, green tariff, seasonality, excess energy from alternative sources are taken into account. Based on the analysis of these and other indicators, applications for the sale and purchase of energy from players in the virtual market are being generated.

**Purpose and objectives of the research.** The purpose of this work is to develop a method for predicting the amount of electricity that can be prepared for implementation on the local market MicroGrid, taking into account the increase or decrease in prices for green tariffs and electricity in the network. At the same time, on the basis of the correlation-regression analysis of time series that are subject to extrapolation processing, a statistical relationship between the parameters is established and the influence of the selected parameters on the value of demand for electric energy in the future for the forecast period is estimated.

To achieve the goal, the following tasks were considered:

- Estimation of the possibility of forecasting excess electricity using regression analysis;
- Development of a method for processing time series of parameters of the modes of MicroGrid using regression analysis;
- Development of data collection, processing and data transmission system;
- Choice and justification of database type;
- Software development for regression analysis based on regression analysis.

**Relationship of work with scientific programs, plans, themes.** The work was carried out at the National Technical University of Ukraine "Kyiv Polytechnic Institute" at the Department of Industrial Electronics in the framework of the initiative research "Improvement of energy efficiency and resource conservation by means of power electronics for the technology of obtaining highly reliable welded joints of heterogeneous materials", No. of state registration 0116U006924.

**The object of the study** is the processes of generating and consuming electric energy in MicroGrid.

**The subject of the study** is the development of methods for evaluating the effectiveness of the modes of operation of MicroGrid on the basis of regression analysis.

**Research methods.** When solving the tasks set in the work, methods of Machine Learning and regression analysis were used. Mathematical and programmatic calculations are performed on a personal computer using Visual Studio 2017 developer package and C # programming language.

**The scientific novelty of the research data is as follows:**

- The theory of research and analysis of operating modes of distributed generation systems has been further developed;
- Based on the regression analysis, a system has been developed to find an assessment of the effective use of electricity for Microgrid, taking into account seasonal parameters and energy consumption data of the system devices.

**The practical value of the results obtained:**

- The experimental layout developed allows you to implement the collection, processing and transfer of data using standard protocols to the database.
- Using the C # programming language, software designed to calculate the excess energy generated on the basis of regression analysis has been developed.

**Personal applicant's fee.** Master's work is a generalization of the results of theoretical and experimental research conducted by the author with the advice of a scientific supervisor. In works published with co-authors, the author in [1] is to search

and analyze information on existing software products for modeling power supply systems MicroGrid, editing, design; In [2], the search for and analysis of information on machine learning techniques for finding the remaining electric energy for further implementation on the local market MicroGrid, editing, design, development of data acquisition and software.

**Publications.** The main provisions and results obtained in the dissertation work were described in the following articles:

- Orlov Yevhenii devices of renewed sources of energy in the system of micro grid and smart house // current problems of modern technologies "collection of abstracts". - Volume 2, v International Scientific and Technical Conference of Young Scientists and Students November 17-18, 2016, Ternopil - 2016. p. 182 -183 // <http://tntu.edu.ua/storage/news/00002831/zbirnyk- tom2.pdf>;
- Orlov Yevhenii estimation of the effective use and sale of electricity from renewable energy sources in the microgrid system // Scientific notes of the Taurian National University in the name of. Vernadsky Series: Technical Sciences, Vol. 29 (68), No. 5, 2018, part 2, Kyiv, 2018. - p. 26-30.

**Structure and volume of the dissertation.** The dissertation consists of introduction, five sections, conclusions, list of used sources from 56 titles. The total volume of dissertation paper is 122 pages, including 90 pages of the main text of 15 drawings and 17 tables.



## Додаток 1

## Лістинг програмного забезпечення

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace Program1
{
    class LinearRegressionProgram
    {
        static void Main(string[] args)
        {
            Console.WriteLine("\n***
Програма оцінки кількості накопиченої
енергії за допомогою регресійного аналізу
***\n");

            int rows = 7;
            int seed = 1;
            Console.WriteLine("***
Отримання даних за тиждень ****");
            double[][] data =
            DummyData(rows, seed);
            Console.WriteLine("\n
Готово- \n");
            Console.WriteLine("Кіл-сть
витрач. елек-ргії за день - День тижня -
Сезон - Залишок:\n");
            NewMethod(data);

            Console.WriteLine("\n***
Створення матриці проектування з даних
***");
            double[][] design =
            Design(data); // 'матриця проектування'
            Console.WriteLine("\n
Готово- \n");
            Console.WriteLine("Матриці
проектування:\n");
            ShowMatrix(design, 2);
            Console.WriteLine("\n***
Пошук коефіцієнтів з використанням
інверсії ****");
            double[] coef = Solve(design);
            // використовуємо матрицю проектування
            Console.WriteLine("\n
Готово- \n");
            Console.WriteLine("Коефіцієнти:\n");
            ;
            ShowVector(coef, 4);
            Console.WriteLine("");
            Console.WriteLine("Обчислення R-
квадрату\n");
            double R2 = RSquared(data,
            coef); // використовувати початкові дані
            Console.WriteLine("R-
квадрат = " + R2.ToString("F4"));

```

```

Console.WriteLine("\nПрогнозування
залишку для");
        Console.WriteLine("Кіл-сть
витрач. елек-ргії за день = 12");
        Console.WriteLine("День
тижня          = 1 (понеділок)");
        Console.WriteLine("Сезон
= 1 (літо)");
        double y = Income(12, 1, 1,
coef); Console.WriteLine("\nПрогнозований
залишок = " + y.ToString("F2"));
        Console.WriteLine("\n***
Кінець програми ***\n");
        Console.ReadLine();
    } // Головка функція
    private static void
NewMethod(double[][] data)
    {
        ShowMatrix(data, 2);
    }
    static double Income(double x1,
double x2, double x3, double[] coef)
    {
        // x1 = Витрати, x2 = день
тижня, x3 = сезон
        double result; // константа
        result = coef[0] + (x1 * coef[1])
+ (x2 * coef[2]) + (x3 * coef[3]);
        return result;
    }

```

```

static double RSquared(double[][]
data, double[] coef)
    {
        // 'коефіцієнт детермінації'
        int rows = data.Length;
        int cols = data[0].Length;
        // 1. обчислити середнє
значення у
        double ySum = 0.0;
        for (int i = 0; i < rows; ++i)
            ySum += data[i][cols - 1]; //
останній стовпець
        double yMean = ySum / rows;
        // 2. сума квадратів залишків і
суми квадратів
        double ssr = 0.0;
        double sst = 0.0;
        double y; // фактичний
залишок
        double predictedY; //
використовуючи coef []
        for (int i = 0; i < rows; ++i)
        {
            y = data[i][cols - 1]; //
отримати актуальний у
            predictedY = coef[0];
            for (int j = 0; j < cols - 1; ++j)
                // j кола даних
                predictedY += coef[j + 1]
* data[i][j]; // обережно
            ssr += (y - predictedY) * (y -
predictedY);

```

```

        sst += (y - yMean) * (y -
yMean);
    }
    if (sst == 0.0)
        throw new Exception("Всі
значення у рівні");
    else
        return 1.0 - (ssr / sst);
}
static double[][] DummyData(int
rows, int seed)
{
    // генерувати фіктивні дані
для лінійної регресії
    double b0 = 9.23;
    double b1 = 0.75;
    double b2 = 1.38;
    double b3 = 0;
    Random rnd = new
Random(seed);
    double[][] result = new
double[rows][];
    for (int i = 0; i < rows; ++i)
        result[i] = new double[4];
    for (int i = 0; i < rows; ++i)
    {
        int ed = rnd.Next(10, 21); //
Випадкові значення від 10 до 21
        int sex = rnd.Next(1, 1); //
Встановлюємо значення енерговитрат за
літній сезон

```

```

        int work = i + 1; // Значення
дня тижня в порядку зростання

        double y = b0 + (b1 * ed) +
(b2 * work) + (b3 * sex);
        y += 10.0 *
rnd.NextDouble() - 5.0; // Помилка -5; +5 %

        result[i][0] = ed;
        result[i][1] = work;
        result[i][2] = sex;
        result[i][3] = y; // Залишок
електроенергії
    }
    return result;
}

static double[][]
Design(double[][] data)
{
    int rows = data.Length;
    int cols = data[0].Length;
    double[][] result =
MatrixCreate(rows, cols + 1);
    for (int i = 0; i < rows; ++i)
        result[i][0] = 1.0;

    for (int i = 0; i < rows; ++i)
        for (int j = 0; j < cols; ++j)
            result[i][j + 1] = data[i][j];
}

```

```

        return result;
    }

    static double[] Solve(double[][]
design)
    {
        // знайти коефіцієнти лінійної
регресії
        // 1. знімаємо X-матрицю і Y-
вектор

        int rows = design.Length;
        int cols = design[0].Length;
        double[][] X =
MatrixCreate(rows, cols - 1);
        double[][] Y =
MatrixCreate(rows, 1); // векторний стовпець
        int j;
        for (int i = 0; i < rows; ++i)
        {
            for (j = 0; j < cols - 1; ++j)
            {
                X[i][j] = design[i][j];
            }
            Y[i][0] = design[i][j]; //
останній стовпець
        }

        // 2. B = inv(Xt * X) * Xt * y
        double[][] Xt =
MatrixTranspose(X);
        double[][] XtX =
MatrixProduct(Xt, X);

```

```

        double[][] inv =
MatrixInverse(XtX);
        double[][] invXt =
MatrixProduct(inv, Xt);
        double[][] mResult =
MatrixProduct(invXt, Y);
        double[] result =
MatrixToVector(mResult);
        return result;
    } // Вирішення

    static void ShowMatrix(double[][]
m, int dec)
    {
        for (int i = 0; i < m.Length; ++i)
        {
            for (int j = 0; j < m[i].Length;
++j) { Console.Write(m[i][j].ToString("F" +
dec) + " ");
            }
            Console.WriteLine("");
        }
    }

    static void ShowVector(double[]
v, int dec)
    {
        for (int i = 0; i < v.Length; ++i)
        Console.Write(v[i].ToString("F" +
dec) + " ");
        Console.WriteLine("");
    }
}

```

```

// Матрична програма

static double[][] MatrixCreate(int
rows, int cols)
    // виділяє / створює матрицю,
    ініціалізовану для всіх 0.0
    // перевірте помилки тут
    double[][] result = new
double[rows][];
    for (int i = 0; i < rows; ++i)
        result[i] = new double[cols];
    return result;
}
// -----

static double[][]
MatrixRandom(int rows, int cols,
double minVal, double maxVal,
int seed)
{
    // повернути матрицю з
    випадковими значеннями
    Random ran = new
Random(seed);
    double[][] result =
MatrixCreate(rows, cols);
    for (int i = 0; i < rows; ++i)
        for (int j = 0; j < cols; ++j)
            result[i][j] = (maxVal -
minVal) *
                ran.NextDouble() +
minVal;

return result;
}

// -----

static double[][]
MatrixLoad(string file, bool header,
char sep)
{
    // завантажити матрицю з
    текстового файлу
    string line = "";
    string[] tokens = null;
    int ct = 0;
    int rows, cols;
    // визначено номер рядків і
    стовпців
    System.IO.FileStream ifs =
new
System.IO.FileStream(file,
System.IO.FileMode.Open);
    System.IO.StreamReader sr =
new
System.IO.StreamReader(ifs);
    while ((line = sr.ReadLine()) !=
null)
    {
        ++ct;
        tokens = line.Split(sep); //
        перевірити
    }
}

```

```

        sr.Close(); ifs.Close();
        if (header == true)                                // -----
            rows = ct - 1;
        else
            rows = ct;
        cols = tokens.Length;
        double[][] result =
MatrixCreate(rows, cols);

        // завантаження
        int i = 0; // індекс рядка
        ifs = new
System.IO.FileStream(file,
System.IO.FileMode.Open);
        sr = new
System.IO.StreamReader(ifs);

        if (header == true)
            line = sr.ReadLine(); //
СПОЖИВАТИ ЗАГОЛОВOK                                // -----
        while ((line = sr.ReadLine()) !=
null)
        {
            tokens = line.Split(sep);
            for (int j = 0; j < cols; ++j)
                result[i][j] =
double.Parse(tokens[j]);
            ++i; // наступний рядок
        }
        sr.Close(); ifs.Close();
        return result;
    }

```

```

        static double[]
MatrixToVector(double[][] matrix)
        {
            // Матриця з одного стовпця
            int rows = matrix.Length;
            int cols = matrix[0].Length;
            if (cols != 1)
                throw new Exception("Bad
matrix");
            double[] result = new
double[rows];
            for (int i = 0; i < rows; ++i)
                result[i] = matrix[i][0];
            return result;
        }
    }
    // -----

    static double[][]
MatrixIdentity(int n)
    {
        // Повернути n x n матрицю
        ідентифікації
        double[][] result =
MatrixCreate(n, n);
        for (int i = 0; i < n; ++i)
            result[i][i] = 1.0;

        return result;
    }

```

```

    }

    // -----

    static string
MatrixAsString(double[][] matrix, int dec)
    {
        string s = "";
        for (int i = 0; i < matrix.Length;
++i)
        {
            for (int j = 0; j <
matrix[i].Length; ++j)
            {
                s +=
matrix[i][j].ToString("F" + dec).PadLeft(8) + "
";

                s += Environment.NewLine;
            }
            return s;
        }

    // -----

    static bool
MatrixAreEqual(double[][] matrixA,
double[][] matrixB, double
epsilon)
    {
        // true, якщо всі значення в
матриці A відповідні значення в матриці
        int aRows = matrixA.Length;
        int aCols = matrixA[0].Length;

        int bRows = matrixB.Length;
        int bCols = matrixB[0].Length;

        if (aCols != bRows)
            throw new Exception("Non-
conformable matrices in MatrixAreEqual");

        for (int i = 0; i < aRows; ++i) //
кожен рядок A і B
            for (int j = 0; j < aCols; ++j)
                // кожен стовбець A і B
                //if
                (matrixA[i][j] != matrixB[i][j])
                    if
                    (Math.Abs(matrixA[i][j] - matrixB[i][j]) >
epsilon)
                        return false;
                    return true;
                }

    // -----

    static double[][]
MatrixProduct(double[][] matrixA, double[][]
matrixB)
    {
        int aRows = matrixA.Length;
        int aCols = matrixA[0].Length;
        int bRows = matrixB.Length;
        int bCols = matrixB[0].Length;

        if (aCols != bRows)
            throw new Exception("Non-
conformable matrices in MatrixProduct");
    }

```

```

        double[][] result =
MatrixCreate(aRows, bCols);

        for (int i = 0; i < aRows; ++i) //
кожен рядок A.
            for (int j = 0; j < bCols; ++j)
// кожен стовбець B
                for (int k = 0; k < aCols;
++k) // вонувати доки k < bRows
                    result[i][j] +=
matrixA[i][k] * matrixB[k][j]; // x += y -> x =
x + y

        return result;
    }

// -----

    static double[]
MatrixVectorProduct(double[][] matrix,
double[] vector)
    {
        // результат множення n x m
матриці на векторний стовбець m x 1 (з
отриманням n x 1 векторного стовпця)
        int mRows = matrix.Length; int
mCols = matrix[0].Length;
        int vRows = vector.Length;
        if (mCols != vRows)

            throw new Exception("Non-
conformable matrix and vector in
MatrixVectorProduct");

        double[] result = new
double[mRows];
        for (int i = 0; i < mRows; ++i)
            for (int j = 0; j < mCols; ++j)
                result[i] += matrix[i][j] *
vector[j];

        return result;
    }

// -----

    static double[][]
MatrixDecompose(double[][] matrix, out int[]
perm,
out int toggle)
    {
        // Doolittle LUP розкладання з
частковим поворотом.
        // returns: result -> L
        // perm містить перестановку
рядів; +1 або -1 (рівний або непарний)
        int rows = matrix.Length;
        int cols = matrix[0].Length;
        if (rows != cols)
            throw new Exception("Non-
square matrix");

        int n = rows; // так зручніше

```



```

double[][] result =
MatrixDuplicate(matrix);

perm = new int[n]; //
встановити результат перестановки рядків
for (int i = 0; i < n; ++i) {
perm[i] = i; }

toggle = 1; // toggle відстежує
заміни рядків

for (int j = 0; j < n - 1; ++j) //
кожен стовпець
{
double colMax =
Math.Abs(result[j][j]);

int pRow = j;

for (int i = j + 1; i < n; ++i) //
читає Matt V :
{
if (Math.Abs(result[i][j]) >
colMax)
{
colMax =
Math.Abs(result[i][j]);

pRow = i;
}
}

if (pRow != j) // якщо
найбільше значення не на осі, обмінюються
рядки
{
double rowPtr =
result[pRow];

result[pRow] = result[j];
result[j] = rowPtr;

int tmp = perm[pRow];
perm[pRow] = perm[j];
perm[j] = tmp;

toggle = -toggle;
}

// -----
-----

if (result[j][j] == 0.0)
{
// знайти рядок для
обміну

int goodRow = -1;
for (int row = j + 1; row <
n; ++row)
{
if (result[row][j] != 0.0)
goodRow = row;
}

if (goodRow == -1)

```

```

        throw new
Exception("Cannot use Doolittle's method");           // -----

        double[] rowPtr =
result[goodRow];
        result[goodRow] =
result[j];
        result[j] = rowPtr;

        int tmp = perm[goodRow];
        perm[goodRow] =
perm[j];
        perm[j] = tmp;

        toggle = -toggle;
    }

    for (int i = j + 1; i < n; ++i)
    {
        result[i][j] /= result[j][j];
        for (int k = j + 1; k < n;
++k)
        {
            result[i][k] -=
result[i][j] * result[j][k];
        }
    }

    return result;
} // MatrixDecompose

static double[][]
MatrixInverse(double[][] matrix)
{
    int n = matrix.Length;
    double[][] result =
MatrixDuplicate(matrix);

    int[] perm;
    int toggle;
    double[][] lum =
MatrixDecompose(matrix, out perm, out
toggle);

    if (lum == null)
        throw new
Exception("Unable to compute inverse");

    double[] b = new double[n];
    for (int i = 0; i < n; ++i)
    {
        for (int j = 0; j < n; ++j)
        {
            if (i == perm[j])
                b[j] = 1.0;
            else
                b[j] = 0.0;
        }
    }

```

```

        double[] x =
HelperSolve(lum, b); // використовувати
розкладання

        for (int j = 0; j < n; ++j)
            result[j][i] = x[j];
    }
    return result;
}

// -----

static double[][]
MatrixTranspose(double[][] matrix)
{
    int rows = matrix.Length;
    int cols = matrix[0].Length;
    double[][] result =
MatrixCreate(cols, rows); // індексування
    for (int i = 0; i < rows; ++i)
    {
        for (int j = 0; j < cols; ++j)
        {
            result[j][i] = matrix[i][j];
        }
    }
    return result;
} // TransposeMatrix

// -----

static double
MatrixDeterminant(double[][] matrix)
{
    int[] perm;
    int toggle;
    double[] lum =
MatrixDecompose(matrix, out perm, out
toggle);

    if (lum == null)
        throw new
Exception("Unable to compute
MatrixDeterminant");

    double result = toggle;
    for (int i = 0; i < lum.Length;
        ++i)
        result *= lum[i][i];
    return result;
}

// -----

static double[]
HelperSolve(double[][] luMatrix, double[] b)
{
    // перед тим, як викликати
цього помічника, перманент b,
використовує perm масив

    // для MatrixDecompose
генерує luMatrix

    int n = luMatrix.Length;
    double[] x = new double[n];
    b.CopyTo(x, 0);

```

```

// виділяє / створює дублікат
матриці
double[][] result =
MatrixCreate(matrix.Length,
matrix[0].Length);
for (int i = 0; i < matrix.Length;
++i) // копіює значення
    for (int j = 0; j <
matrix[i].Length; ++j)
        result[i][j] = matrix[i][j];
return result;
}

// -----
-----

static double[][]
ExtractLower(double[][] matrix)
{
    int rows = matrix.Length; int
cols = matrix[0].Length;
    double[][] result =
MatrixCreate(rows, cols);
    for (int i = 0; i < rows; ++i)
    {
        for (int j = 0; j < cols; ++j)
        {
            if (i == j)
                result[i][j] = 1.0;
            else if (i > j)
                result[i][j] =
matrix[i][j];

```

```

for (int i = 1; i < n; ++i)
{
    double sum = x[i];
    for (int j = 0; j < i; ++j)
        sum -= luMatrix[i][j] *
x[j];

    x[i] = sum;
}
if(luMatrix[n - 1][n - 1] == 0)
{
    luMatrix[n - 1][n - 1] = 1;
}
x[n - 1] /= luMatrix[n - 1][n - 1];
for (int i = n - 2; i >= 0; --i)
{
    double sum = x[i];
    for (int j = i + 1; j < n; ++j)
        sum -= luMatrix[i][j] *
x[j];

    x[i] = sum / luMatrix[i][i];
}

return x;
}

// -----

static double[][]
MatrixDuplicate(double[][] matrix)
{

```

```

    }
    }
    return result;
}

static double[][]
ExtractUpper(double[][] matrix)
{
    int rows = matrix.Length; int
cols = matrix[0].Length;
    double[][] result =
MatrixCreate(rows, cols);
    for (int i = 0; i < rows; ++i)
    {
        for (int j = 0; j < cols; ++j)
        {
            if (i <= j)
                result[i][j] =
matrix[i][j];
        }
    }
    return result;
}

// -----
-----

static double[][]
PermArrayToMatrix(int[] perm)
{
    // перетворює масив Doolittle
perm в відповідну матрицю perm

    int n = perm.Length;
    double[][] result =
MatrixCreate(n, n);
    for (int i = 0; i < n; ++i)
        result[i][perm[i]] = 1.0;
    return result;
}

static double[][]
UnPermute(double[][] luProduct, int[] perm)
{
    double[][] result =
MatrixDuplicate(luProduct);

    int[] unperm = new
int[perm.Length];
    for (int i = 0; i < perm.Length;
        ++i)
        unperm[perm[i]] = i;

    for (int r = 0; r <
luProduct.Length; ++r)
        result[r] =
luProduct[unperm[r]];

    return result;
} // UnPermute
} // Program

```